Lecture Notes

Wolfgang Förstner

May 7, 2024

"Whatever it takes to finish things, finish.
You will learn more from a glorious failure
than you ever will from something you never finished."
Neil Gaiman

"Unlike the novel, a short story may be, for all purposes, essential."
Jorge Luis Borges

Preamble

This is a collection of notes I wrote for preparing lectures, for publications or for communicating with collegues on interesting topics. They partially go back more than two decades. May be it is useful to make them public, though the relevance of each individual note, of cource, has to be evaluated by the reader.

The notes are of different nature: some are just proofs of lemmata, some are collections of ideas on a certain topic as a basis for a discussion, some are basis for supplements of publications, some address problems I encountered and are of more conceptual nature, finally, some notes are used for lectures.

I left the texts more or less as I wrote them, translated them if the original was written in German. However, in order to comply with some standards in machine learning, I homogeneized the notation following some statistical conventions: observations are denoted with $\boldsymbol{y}$, unknown parameters with $\boldsymbol{\theta}$. I also augmented each note with a preface containing a short summary and indicating the main motivation for including it in this collection. Some of the notes refer to code, mostly written in MATLAB, which I also make public.

I did not recheck all derivations, i.e., it is up to the reader to check the details, especially in notes, which mainly have been meant to explore an idea.

Part I collects lecture notes. The technical notes in the next parts are sorted according to the following topics I was interested in the last decades

II Statistics and Estimation Theory

III Estimation and Uncertainty of Geometric Entities

IV Image Analysis

V Geometry

VI Bundle Adjustment and Surface Reconstruction

VII Miscellaneous

I did not include an index, since it appears easier to use the searching capablities of todays electronic readers.

Some notes are still under construction and will be updated when reaching the next level of maturity. The same holds for supproting software.

I hope the reader finds this collection interesting.

Wolfgang Förstner, Bonn
May 7, 2024

Contents

Part I

Lecturenotes

1 Information Theory

These lecture notes on Information Theory cover basic concepts of self-information and entropy for characterizing the coding of probabilistic models and measures for comparing theoretical or empirical densities, both for discrete and continuous random variables. We address the principle of maximum entropy for choosing densities under uncertainty and relate the task of model selection to robust estimation.

1.1 Preface

Information theory was the starting point for my lectures on pattern recognition. Together with Konrad Schindler, we augmented the notes for a book section. The motivation for writing such a text goes back to the PhD of George Vosselman Vosselman (1992), who intensively used information theory for matching relational structures and used the self-information as measure for the quality of correspondences, an idea later used by Kolbe (2000) identifying stochastical models of buildings with the ability to code them. Also the early work by Georgeff and Wallace (1984) was stimulating to find a relation of model selection via description length to robust estimation.

1.2 Introduction

The theory of coding and information is closely related to probability theory. Often the specification of a probability that an event occurs can be replaced by the specification of the gain in information when being told an event. For the purposes of this book it regularly is an alternative tool to express and manipulate the uncertainty of observations, models and predictions. We assume a basic understanding of probability theory[1], as introduced in textbooks like Papoulis (1991); Bishop (2006) or in Förstner and Wrobel (2016, Chap. 2), which gives a compact summary of the most important results and the corresponding notation. A good textbook on information theory is Cover and Thomas (1991).

[1] Discrete probability distribution; random variables; continuous probability densities; cumulative densities; conditional, joint and marginal distributions; Bayes' rule; expectation, variance and covariance; moments of a distribution; functions of random variables.

An introductory example

To motivate the use of information theory, we start with an example. Imagine that a number of points have been detected in an image, e.g., small bits of forest floor visible between the tree crowns (see Fig. 1.1); and that our task is to determine whether they form a straight line, e.g., a road traversing the forest or they are just random. We must take a decision under uncertainty: the points will not lie exactly on the centreline of the putative road, and some points might be outliers on small clearings or the like. Also, we may not know the width of the putative road.

Is it more likely that most points form a road and the rest are outliers (model A), that a few points form a road with small residuals and there are more outliers (model B), that there is no road and some points are roughly collinear by chance (model C), or even that the points only roughly describe a straight line and we have only 6 outliers (model D)?

To guide our choice, we rely on a fundamental assumption of data interpretation (and science in general), namely that simpler models are a priori more likely[2]. As we will see in the following chapter, one way to formalize the prior probability of a model, in our case a straight line with a given standard deviation, is through its information-theoretic *coding length*: if the fitting residuals are small, then transmitting the parameters of the line, the nominal positions of the points on the line and the residuals is more efficient than transmitting the point coordinates.

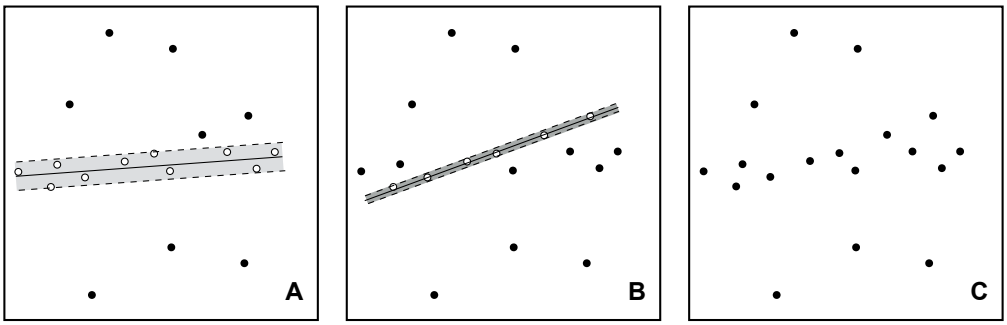


Figure 1.1: Model selection under uncertainty: the same set of points can be explained by three different models. **Left:** Some points lie on a board road. **Middle:** Some (other) points lie on a narrow road. **Right:** Some points are random. Information theory provides tools to compare the three solutions. Later, Tab. 1.1 we will see that the best model appears to be model B, assuming the points lie in a range of $R = 256$, and standard deviations $\sigma_A = 5.8$ and $\sigma_B = 1.6$

As illustrated by the example, the formal theory of information originates from the need to analyse the transmission of messages over (noisy) channels. A message is coded, sent through the channel, and must be decoded by the receiver. More messages can be sent if a good coding scheme is selected, which *on average* leads to shorter messages. Since the content of individual messages is not known, we need to know the expected coding length[3] which depends on the statistical structure of the transmitted signals. [4] Most obviously, only what is uncertain must be transmitted, facts already known to the receiver carry no information: in our example, we need not include in the message the y-coordinate of a point, which is assumed to be on a line, since it can be computed from the line parameters and its x-coordinate. More generally, the concepts of *information* and *probability* are closely related: improbable outcomes carry more information – informally,

[2] Actually, scientists - and also most humans - tend to choose simple models. Since the true model underlying reality is not known, the choice of a model is free, so models can be chosen just to serve a purpose, defined by the user of a model. Therefore, simpler models usually appear, i.e., are chosen more likely. *This discussion needs to appear in the introduction to the book.*

[3] Not the 'channel capacity'

[4] This was first established in the ground-breaking work of C. E. Shannon (1949).

we can think of information as a measure of surprise.

1.3 Information, Entropy and Probability

The information $\mathrm{I}(A)$ of message A should measure how much knowledge is added by receiving A. It should thus have the following properties:

1. It should never be negative,
$$\mathrm{I}(A) \geq 0 \tag{1.1}$$
2. It should be 0 if and only if its argument has no uncertainty,
$$\mathrm{I}(A) = 0 \quad \Leftrightarrow \quad \mathrm{P}(A) = 1 \tag{1.2}$$
3. For two independent messages A and B with $\mathrm{P}(A,B) = \mathrm{P}(A)\mathrm{P}(B)$, the total information should be the sum of their individual information values.
$$\mathrm{P}(A,B) = \mathrm{P}(A)\mathrm{P}(B) \quad \rightarrow \mathrm{I}(A,B) = \mathrm{I}(A) + \mathrm{I}(B) \tag{1.3}$$

To fulfil these conditions, the information of A must be proportional to the negative logarithm of its probability. Logarithms with different base differ by a constant factor, i.e., the base defines the unit of information. We will use base 2, corresponding to a unit of 1 bit. Another common choice is the *nat*ural logarithm with base e, the units of which are sometimes called nats.

1.3.1 Information

We have the following definition of the information of a message, which can be generalized to messages given under a known condition and to the mutual information of two messages.

Definition 1.3.1: ***Information.*** The information $\mathrm{I}(A)$ of a specific message A which has probability $\mathrm{P}(A) = \mathrm{P}(\underline{A} = A)$ is *self-information*

$$\mathrm{I}(A) = -\mathrm{lb}\,\mathrm{P}(A) \geq 0\,. \tag{1.4}$$

The information is non-negative. We also call it "self-information" in order to distinguish it from its expected value, the entropy, see below Sect. 1.3.2. ◇

Similar to probabilities $\mathrm{P}(A) = \mathrm{P}(\underline{A} = A)$ also the information of a message $\underline{A} = A$ depends on the context, say C. So, we always could write $\mathrm{P}(\underline{A} = A \mid C)$ instead of $\mathrm{P}(\underline{A} = A)$, and now, $\mathrm{I}(\underline{A} = A \mid C)$ instead $\mathrm{I}(\underline{A} = A)$. We omit writing the context as a condition, and assume it is clear when specifying probabilities or information.

Observe, we distinguish between the random variable $\underline{A}$ which describes the possible outcomes of an experiment and a sample or realization A from this random variable, or outcome from this experiment. The sample, i.e., the specific outcome, A as such is not uncertain, just the possible outcomes of the experiment[5].

Example 1.3.1: ***Information of one grey value.*** If we assume that all intensities of a grey level image are equally likely, and lie in the range $[0, ..., 255]$ then a randomly chosen pixel in such an 8-bit image has intensity $g = 77$ with probability $\frac{1}{256}$. The observation $A = \{\texttt{Grey value at pixel } \boldsymbol{x} \texttt{ is } 77\}$ has $\mathrm{I}(A) = -\mathrm{lb}\,\frac{1}{256} = 8$ bits of information.

The example immediately shows the relation between information and coding length, discussed below: to store or transmit the intensity of the pixel, we need 8 bits. Without further information about the likelihood of different intensity patterns, it takes 8 bit per pixel to encode a gray-value image. ◇

Definition 1.3.2: ***Conditional Information.*** For two messages A and B, the conditional information of A, given that we have already know, i.e., decoded B, is *conditional self-information*

[5] see Savage (1972): "Nonetheless, definitive observations do not play an important part in statistical theory, precisely because statistics is mainly concerned with uncertainty, and there is no uncertainty once an observation definitive for the context at hand has been made.".

$$\mathrm{I}(A|B) = -\mathrm{lb}\, P(A|B) \geq 0\,, \tag{1.5}$$

again measured in [bit]. It is a measure of the degree of surprise to hear A when B is already known. ◇

We obviously have

$$\mathrm{I}(A|B) = -\mathrm{lb}\,\frac{\mathrm{P}(A,B)}{\mathrm{P}(B)} = \mathrm{I}(A,B) - \mathrm{I}(B)\,, \tag{1.6}$$

Hence, the surprise $\mathrm{I}(A|B)$ when receiving message A when B is known, is smaller by $\mathrm{I}(B)$ compared to the surprise about the combined message $\mathrm{I}(A,B)$.

Example 1.3.2: ***Information of a grey value in an image with weak contrast.*** Suppose we already know that an image has weak contrast thus message $B =$ `Grey value at` $\boldsymbol{x}$ `is in the range` $[60\ldots91]$. Then the conditional information of the message $A = \{$`Grey value at pixel` $\boldsymbol{x}$ `is 77`$\}$ given B is $\mathrm{I}(A|B) = -\mathrm{lb}\, P(A|B) = -\mathrm{lb}\,\frac{1}{32} = 5$ bit. Message A now carries less information, because B has already ruled out most of the possible grey values. ◇

When making predictions from observed data, we are often interested in the information that one statement A holds about another statement B: if the observations have a certain value x, what knowledge does that give us about the unknowns of interest y?

Definition 1.3.3: ***Mutual Self-Information.*** The mutual self-information $\mathrm{I}(A;B)$ of two messages is

mutual self-information

$$\mathrm{I}(A;B) = \mathrm{I}(A) + \mathrm{I}(B) - \mathrm{I}(A,B) \in (-\infty, +\infty)\,. \tag{1.7}$$

It is symmetric w.r.t. the two messages and may be positive or negative. ◇

Remark: Observe, the prevalent terminology is inconsistent: as we will see below, the term *mutual information* is widely used for another quantity, which should in fact be called *mutual entropy*. To avoid confusion, we will always refer to $\mathrm{I}(A;B)$ as *mutual self-information*. ◇

The definition immediately leads to the relation

$$\mathrm{I}(A;B) = \mathrm{I}(A) - \mathrm{I}(A|B) = \mathrm{I}(B) - \mathrm{I}(B|A)\,, \tag{1.8}$$

and thus

$$\mathrm{I}(A|B) = \mathrm{I}(A) - \mathrm{I}(A;B) \quad \text{and} \quad \mathrm{I}(B|A) = \mathrm{I}(B) - \mathrm{I}(A;B)\,. \tag{1.9}$$

Hence, if the mutual self-information is positive, then receiving message B reduces the initial surprise about message A. Otherwise, namely if A and B contradict to some extent, the mutual self-information is negative, and the receiving message B increases the surprise about message A.

Example 1.3.3: ***Mutual self-information of neighbouring pixels.*** An image spans the whole 8-bit range of gray-values, such that again the message $A = \{g(i,j) = 77\}$ has $\mathrm{I}(A) = 8$ [bit]. However, the values vary smoothly, e.g., due to out-of-focus blur, such that neighbouring pixels differ by at most $[-4\ldots+3]$ grey values. We now observe $B = \{g(i,j+1) = 79\}$. This measurement contains quite a lot of information also about A, because of the blur: $g(i,j)$ must be between 75 and 82, hence $\mathrm{P}(A|B) = \frac{1}{8}$. Accordingly, $\mathrm{I}(A;B) = \mathrm{I}(A) - \mathrm{I}(A|B) = 8 - 3 = 5$ [bit]. ◇

Example 1.3.4: ***Negative mutual self-information.*** Let us have two messages: $A =$ `It is raining` and $B =$ `I have opened my umbrella`. Then the two messages A and $\neg B =$ `I have not open my umbrella` have negative mutual-self-information, since after knowing A the surprise when receiving the message $\neg B$ is larger than when receiving the message B. ◇

Exercise 1.1

Note the close connection to Bayes' law: our prior belief $\mathrm{P}(B)$ is updated by the observed evidence A via

$$\mathrm{P}(B|A) = \frac{\mathrm{P}(A|B)}{\mathrm{P}(A)}\mathrm{P}(B) = \frac{\mathrm{P}(A,B)}{\mathrm{P}(A)\mathrm{P}(B)}\mathrm{P}(B)\,. \tag{1.10}$$

Analogously to the mutual information, we name the (inverse) update factor

$$\mathrm{P}(A;B) := \frac{\mathrm{P}(A)\mathrm{P}(B)}{\mathrm{P}(A,B)} \geq 0 \quad \text{and} \quad \mathrm{I}(A;B) = -\mathrm{lb}\,\mathrm{P}(A;B) \tag{1.11}$$

Taking negative logarithms, this is equivalent to 1.7 and directly leads to 1.9. Observe, the factor $\mathrm{P}(A|B)/\mathrm{P}(A) = 1/\mathrm{P}(A;B)$ can be smaller or larger than 1, indicating that the observation A may cause the posterior probability $\mathrm{P}(B|A)$ to be smaller or larger than the prior probability $\mathrm{P}(B)$.

If there is no correlation between an observation and the quantity of interest, the mutual information vanishes,

$$\mathrm{P}(A,B) = \mathrm{P}(A)\mathrm{P}(B) \quad \Leftrightarrow \quad \mathrm{I}(A;B) = 0\,. \tag{1.12}$$

Visually, one can imagine the mutual information as the "degree of overlap" between two messages.

1.3.2 Entropy

So far, we have discussed the "self-information" of a single, specific message. More often, we are interested in the *expected* information, or in other words the mean information content of messages, when only their probability distribution is known.

Definition 1.3.4: ***Entropy.*** Let $\underline{A}$ be a (discrete) random variable, which can take on values $(A_1, ..., A_n, ..., A_N)$. Then its entropy, $\mathrm{H}(\underline{A})$, is the expected information [6] *entropy*

$$\mathrm{H}(\underline{A}) = \mathbb{E}(\underline{I}(A)) = -\sum_{n=1}^{N} P(A_n)\,\mathrm{lb}\,P(A_n) \geq 0\,. \tag{1.13}$$

◇

According to Hjalmars (1977) the letter H for the entropy is likely to result from the Greek letter η, whose capital form is not distinguishable from the Latin letter H.

Furthermore, the argument of the entropy is a random variable, as for the expectation for a random variable. We will make this explicit within this introductory section, but *simplify notation later*, by omitting the underline for the random variable.

Example 1.3.5: ***Entropy of binary image.*** The pixel values of a binary image are distributed according to $\mathrm{P}(\underline{A} = 0) = p$, respectively $\mathrm{P}(\underline{A} = 1) = 1 - p$. Then the entropy of the image is meant to be the expected value of the information, we obtain, when being told the value of one of its grey values: *entropy of binary variable*

$$\mathrm{H}(p) := \mathrm{H}(\underline{A}) = p\,\mathrm{lb}\,p + (1-p)\,\mathrm{lb}\,(1-p)\,. \tag{1.14}$$

The entropy depends on the likelihood that a pixel is 0 or 1, as shown in Figure 1.2. It peaks at $p = 1/2$, where it becomes 1: the uncertainty about the image reaches its maximum when black and white pixels are equally likely. In that case, 1 bit/pixel is needed to encode the image, whereas the uncertainty about each individual pixel is lower if, statistically, a large majority is white (or black). In the extreme cases $p = 0$ or $p = 1$ we know that the entire image is white (respectively, black), hence the entropy is $\mathrm{H}(\underline{A}) = 0$. ◇ Exercise 1.2

Theorem 1.3.1: ***Entropy as Minimal Coding Length.*** A code for a sequence $(A_1, ..., A_n)$ of independent samples of the random variable $\underline{A}$ cannot have less than $\mathrm{H}(\underline{A})$ bits per sample, see (Shannon and Weaver, 1949) *coding theorem*

The entropy thus is a lower bound for the (average, non-integer) number of bits needed to transmit each realisation of a random variable. As we will see, this central finding of coding theory is also useful to select and compare suitable data representations, via the analogy from Sect. 1.2.

Example 1.3.6: ***Binary noise image.*** A binary noise image therefore cannot be coded with less than 1 [bit/pixel]. ◇

The definitions of conditional and mutual information can be carried over to entropy.

Definition 1.3.5: ***Conditional Entropy.*** The conditional entropy $\mathrm{H}(\underline{A}|\underline{B})$ is the expectation of the conditional information $\underline{\mathrm{I}}(A|B) = \mathrm{I}(\underline{A}|B)$:

$$\mathrm{H}(\underline{A}|B) = \mathbb{E}(\underline{\mathrm{I}}(A|B)) = -\sum_{n=1}^{N} \mathrm{P}(A_n|B)\,\mathrm{lb}\,\mathrm{P}(A_n|B) \geq 0\,. \tag{1.15}$$

[6] Note, formally the definition implies that the information $\underline{\mathrm{I}}(A) = I(\underline{A})$ now is a function of the random variable $\underline{A}$, such that the input to the expectation is a random variable.

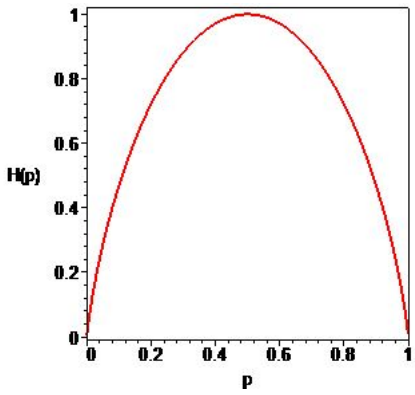


Figure 1.2: Entropy of a binary variable $\underline{A}$ as a function of the probability $p = \mathrm{P}(\underline{A})$

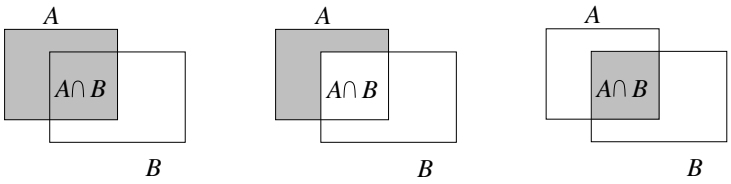


Figure 1.3: Visual illustration of entropy. **Left:** the gray area depicts the expected information $\mathrm{H}(\underline{A})$ obtaining from a sample A of $\underline{A}$ – larger area means greater expected information (surprise about a specific outcome). **Center:** the expected additional information of of A after knowing the outcome B for $\underline{B}$, i.e., $\underline{A} \setminus B$ is $\mathrm{H}(\underline{A}|B)$. **Right:** the mutual information $\mathrm{H}(\underline{A};\underline{B}) = \mathrm{H}(\underline{A}) - \mathrm{H}(\underline{A}|B)$ measures how much the answer to B on an average influences, i.e., diminuishes $\mathrm{H}(\underline{A})$ and vice versa

Note that, like the conditional probability, it is not symmetric, $\mathrm{H}(\underline{A}|B) \neq \mathrm{H}(\underline{B}|A)$. ◇

Definition 1.3.6: ***Mutual Entropy (Mutual Information).*** The mutual entropy $\mathrm{H}(\underline{A};\underline{B})$ is the expectation of the mutual self-information,

$$\mathrm{H}(\underline{A};\underline{B}) = \mathbb{E}(\underline{\mathrm{I}}(A;B)) = \sum_A \sum_B \mathrm{P}(A,B)\mathrm{I}(A;B) = -\sum_A \sum_B \mathrm{P}(A,B)\log \mathrm{P}(A;B) \geq 0\,. \tag{1.16}$$

It is symmetric w.r.t. the two variables and positive. ◇

Remark: Once again, we point out that the prevalent terminology is inconsistent: $\mathrm{H}(\underline{A};\underline{B})$ is commonly referred to as *mutual information*. We will follow this convention to be consistent with other literature, and call $\mathrm{I}(A;B)$ *mutual self-information* throughout this book. ◇

Typically, the mutual information is computed via one of the following expressions

$$\mathrm{H}(A;B) = \mathrm{H}(A) - \mathrm{H}(A|B) = \mathrm{H}(B) - \mathrm{H}(B|A) = \mathrm{H}(A) + \mathrm{H}(B) - \mathrm{H}(A,B)\,. \tag{1.17}$$

Fig. 1.3 shows the relation between the entropy, the conditional entropy and the mutual entropy. Observe, the overlap between the areas for $\underline{A}$ and $\underline{B}$ is non-negative, indicating the expected mutual information $\mathrm{H}(\underline{A};\underline{B})$ is not negative. This is in contrast to the mutual self-information $\mathrm{I}(A;B)$ of two samples A and B, which may be positive, i.e., in case they support each other, or negative, i.e., in case they contradict each other.

1.4 Kullback-Leibler Divergence

In some cases it is useful to quantify "how different" two probability distributions $\mathrm{P}(A)$ and $\mathrm{P}(B)$ are. The interpretation of entropy as minimal coding length suggests the following procedure: encode samples from $\mathrm{P}(A)$ with the optimal code for $\mathrm{P}(B)$, and measure how much additional coding length is required, compared to the optimal code for $\mathrm{P}(A)$. This

leads to two important concepts, the cross-entropy and the divergence of two probabilities, proposed by Kullback and Leibler (1951).

Definition 1.4.7: ***Cross-Entropy.*** The cross-entropy is defined as the expected coding length needed to encode a message A, following $\mathrm{P}(A)$, but using the optimal code for $\mathrm{P}(B)$,

$$\mathrm{H}_A(\underline{B}) = \mathbb{E}_A(\underline{\mathrm{I}}(B)) = -\sum_{n=1}^{N} \mathrm{P}(A_n)\,\mathrm{lb}\,\mathrm{P}(B_n)\,. \tag{1.18}$$

◇

The cross entropy specializes to the entropy $\mathrm{H}_A(\underline{A}) = \mathrm{H}(\underline{A})$ if $P(A) = P(B)$. What is most essential: due to the coding theorem on p. 11 the cross entropy is always larger than the entropy: $\mathrm{H}_A(\underline{B}) \geq \mathrm{H}(\underline{A})$. This is the motivation for the following concept, the Kullback-Leibler divergence. *KL-divergence*

Definition 1.4.8: ***Kullback-Leibler Divergence.*** The KL-divergence is the expected increase in coding length that one incurs by encoding a message A with the code optimised for $\mathrm{P}(B)$,

$$\mathrm{D}_{\mathrm{KL}}\big(\mathrm{P}(A)\|\mathrm{P}(B)\big) = \mathrm{H}_A(B) - \mathrm{H}_A(A) = \sum_{n=1}^{N} \mathrm{P}(A_n)\,\mathrm{lb}\,\frac{\mathrm{P}(A_n)}{\mathrm{P}(B_n)} \geq 0 \tag{1.19}$$

◇

The KL-divergence is 0 if $\mathrm{P}(A)$ and $\mathrm{P}(B)$ are the same, and positive otherwise. It is not a metric, since it is not symmetric and the triangle inequality is not fulfilled. If needed, a symmetric, non-negative measure can be obtained via Exercise 1.3

$$\mathrm{D}_{\mathrm{KL}}^{\mathrm{sym}} = \mathrm{D}_{\mathrm{KL}}\big(\mathrm{P}(A)\|\mathrm{P}(B)\big) + \mathrm{D}_{\mathrm{KL}}\big(\mathrm{P}(B)\|\mathrm{P}(A)\big)\,. \tag{1.20}$$

Note that the mutual information can be interpreted as the KL-divergence between the messages' joint probability $\mathrm{P}(A,B)$ and the product of the marginals,

$$\mathrm{H}(\underline{A};\underline{B}) = \mathbb{E}_{(A,B)}\left(\underline{\mathrm{I}}(A) + \underline{\mathrm{I}}(B) - \underline{\mathrm{I}}(A,B)\right) = \mathrm{D}_{\mathrm{KL}}\big(\mathrm{P}(A,B)\|\mathrm{P}(A)\mathrm{P}(B)\big) \geq 0\,. \tag{1.21}$$

Intuitively, this makes sense: if the two variables are independent, $\mathrm{P}(A,B) \equiv \mathrm{P}(A|B)\mathrm{P}(B) = \mathrm{P}(A)\mathrm{P}(B)$, the messages do not get longer if we use the product of the marginals $\mathrm{P}(A)$ and $\mathrm{P}(B)$ instead of the joint distribution for encoding. But, as can be seen from 1.17: if the mutual information grows, the conditional entropy $\mathrm{H}(A|B)$ shrinks accordingly, and encoding with $\mathrm{P}(A)$, which requires $\mathrm{H}(A)$ bits becomes increasingly wasteful.

Example 1.4.7: ***Coding an Unfair Dice.*** We compare the outcomes of throwing a fair (A) and an unfair dice (B). A fair dice yields the numbers one to six with equal probability $\mathrm{P}(A) = 1/6$. The message, that the result of a throw is $\underline{A} = 6$ has $-\mathrm{lb}\,1/6 = 2.585$ bit; this at the same time is the minimum average coding length $\mathrm{H}(A)$ for the outcome of a throw.

Let us now assume we code a sequence of independent throws differently, namely by coding sixtupels of outcomes $(x_1, ..., x_6)$ by determining the number $t = \sum_{i=1}^{6} x_i 6^{i-1} \leq 46\,656 < 2^{16}$ and representing the outcome of this triple with an 16-bit number in $\{0, 65536\}$, and we need $16/6 = 2.667$ bits, which is slightly worse than the optimum $H_A(A) = 2.585$.

If the dice is not fair, we need to expect that on an average the coding length is shorter. For example, if the probabilities for throwing the number one to six are $[1/10, 1/10, 1/10, 1/10, 1/10, 1/2]$, we have $\mathrm{H}(B) = 2.161$. Hence the increase in expected coding length when coding the unfair dice, assuming it is fair is

$$\mathrm{D}_{\mathrm{KL}}(P(B)||P(A)) = H_B(A) - H_B(B) = 2.5850 - 2.1561 = 0.4240\,[\mathrm{bits}]\,, \tag{1.22}$$

while when coding the fair dice using the optimal code for the unfair dice the increase in expected coding length is

$$\mathrm{D}_{\mathrm{KL}}(P(A)||P(B)) = H_A(B) - H_A(A) = 2.9349 - 2.5850 = 0.3500\,[\mathrm{bits}]\,. \tag{1.23}$$

◇

1.5 Differential Information

We will now generalise the concepts of information and entropy to continuous random variables, say $\underline{x}$. To do so, we face the difficulty that, for any single outcome, the probability is $\mathrm{p}(x_i) = 0$ and, accordingly, the information is $\mathrm{I}(x_i) = \infty$.

If we discretize the random variable with step length Δx, then in a first-order approximation we have

$$\mathrm{P}(\underline{x} \in [x, x + \Delta x]) = p(x)\Delta x \tag{1.24}$$

Taking the (negative) logarithm, in order to obtain the information, we formally get

$$\mathrm{I}(\underline{x} \in [x, x + \Delta x]) = -\mathrm{lb}\,\mathrm{p}(x) - \mathrm{lb}\,\Delta x \tag{1.25}$$

The first term formally is the same as for the discrete case. The second term goes to infinity as $\Delta x \to 0$, but does not depend on $\mathrm{p}(x)$.

Remark: We need to be aware, that generally, if the random variable, which is assumed to be unitless, is taken as the real number of a measurement without its unit, say [m], then – without mentioning – the density $\mathrm{p}(x)$ depends on that unit. E.g. if a length is measured in [m] its density might be a Gaussian $\mathrm{p}(x) = \mathrm{g}(x; 1, 0.01^2)$, saying the mean length is 1 [m] and the standard deviation 0.01 [m], while if the same length is measured in [cm] we obtain $\mathrm{p}(x) = \mathrm{g}(x; 100, 1^2)$ with the mean length 100 [cm] and the standard deviation 1 [cm], for the user the same meaning, but with different probabilistic description. The definition of the self-information of a continuous variable, thus, needs to handle both properties: (1) the possible dependency of $\mathrm{p}(x)$ on the unit of the underlying measurements and (2) the dependency of the free interval Δx. ◇

The conventional definition of differential self-information just takes the logarithmic term, (1) not making the possible units of the underlying measurements explicit, and (2) neglecting the second term with Δx:

Definition 1.5.9: ***Differential Self-Information.*** The differential self-information of a continuous random variable $\underline{x}$ with probability density $\mathrm{p}(x)$ is

$$\mathrm{I}(x) = -\mathrm{lb}\,\mathrm{p}(x) \tag{1.26}$$

◇

When comparing the information of continuous features of measurements having a unit, we need to *fix the discretization*, thus work with discrete variables.

Therefore, the differential self-information is not invariant to scaling. Hence, changing the unit of the variable x changes the differential information. It changes according to

$$\mathrm{I}(ax) = \mathrm{I}(x) + \mathrm{lb}\,a \tag{1.27}$$

For vector-valued random variables, the relation reads

$$\mathrm{I}(A\boldsymbol{x}) = \mathrm{I}(\boldsymbol{x}) + \mathrm{lb}\,|A| \tag{1.28}$$

Example 1.5.8: ***Information of Uniform Distribution.*** A uniformly distributed random variable $\underline{x} \sim \mathcal{U}(a, b)$ has probability $\mathrm{p}(x) = 1/(b-a)$ everywhere in $[a, b]$, and differential self-information

$$\mathrm{I}_U(x|a, b) = -\mathrm{lb}\,\frac{1}{b-a} = \mathrm{lb}\,(b-a) \tag{1.29}$$

A larger interval $[a, b]$ requires longer codes. The code length increases by 1 bit every time the interval is doubled. ◇

Example 1.5.9: ***Information of Gaussian Distribution.*** A Gaussian random variable $\underline{x} \sim \mathcal{N}(\mu, \sigma)$ has probability density $\mathrm{p}(x) = \mathrm{g}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$, and differential self-information

$$I_N(x|\mu, \sigma) = \frac{\mathrm{lb}\,\mathrm{e}}{2}\left(\frac{x-\mu}{\sigma}\right)^2 + \frac{1}{2}\mathrm{lb}\,2\pi\sigma^2 \quad [\mathrm{bit}] \tag{1.30}$$

using the relation $\mathrm{lb}\,(x) = \ln(x)/\ln(2)$. ◇

Further quantities are defined in a similar manner:

Definition 1.5.10: ***Differential Entropy.*** The differential entropy of a continuous random variable $\underline{x}$ with density $\mathrm{p}(x)$ is

$$\mathrm{h}(\underline{x}) = \mathbb{E}(\underline{\mathrm{I}}(x)) = -\int_{x=-\infty}^{\infty} \mathrm{p}(x)\mathrm{lb}\,\mathrm{p}(x) \tag{1.31}$$

◇

Example The entropies for the above examples are

$$\mathrm{h}_U(x|a,b) = \mathrm{lb}\,(b-a) \quad [\mathrm{bit}] \tag{1.32}$$

$$\mathrm{h}_N(x|\mu,\sigma) = \frac{1}{2}\mathrm{lb}\,2\pi\mathrm{e}\sigma^2 \quad [\mathrm{bit}] \tag{1.33}$$

Definition 1.5.11: ***Differential Conditional Self-Information.*** The differential conditional self-information of a continuous random variable $\underline{x}$ with density $\mathrm{p}(x|y)$ is

$$\mathrm{I}(x|y) = -\mathrm{lb}\,\mathrm{p}(x|y)\,. \tag{1.34}$$

◇

Definition 1.5.12: ***Differential Conditional Entropy.*** The differential conditional entropy of a continuous random variable $\underline{x}$ with density $\mathrm{p}(x|y)$ is

$$\mathrm{h}(\underline{x}|\underline{y}) = \mathbb{E}(\underline{\mathrm{I}}(x|y)) = -\int_{x=-\infty}^{\infty} \mathrm{p}(x|y)\mathrm{lb}\,\mathrm{p}(x|y) \tag{1.35}$$

◇

Definition 1.5.13: ***Differential Mutual Self-information.*** The differential mutual self-information of two continuous random variables $\underline{x}$ and $\underline{y}$ with joint density $\mathrm{p}(x,y)$ is

$$\mathrm{I}(x;y) = \mathrm{I}(x) - \mathrm{I}(x|y) = \mathrm{I}(y) - \mathrm{I}(y|x) \tag{1.36}$$

◇

Definition 1.5.14: ***Differential Mutual Entropy (Differential Mutual Information).*** The differential mutual entropy of two continuous random variables $\underline{x}$ and $\underline{y}$ with joint density $\mathrm{p}(x,y)$ is

$$\mathrm{h}(\underline{x};\underline{y}) = \mathrm{h}(\underline{x}) - \mathrm{h}(\underline{x}|\underline{y}) = \mathrm{h}(\underline{y}) - \mathrm{h}(\underline{y}|\underline{x}) = \mathbb{E}(\underline{\mathrm{I}}(x,y)) \tag{1.37}$$

◇

Again, the term *mutual information* is prevalent in the literature, being used for discrete and continuous random variables.

Example: Two Gaussian random variables, which are correlated with correlation coefficient ρ_{xy} have mutual differential information

$$\mathrm{h}(\underline{x};\underline{y}) = \frac{1}{2}\mathrm{lb}\,\frac{1}{1-\rho_{xy}^2} \quad [\mathrm{bit}] \tag{1.38}$$

Their mutual differential information depends only on the correlation coefficient ρ_{xy}. The mutual information only is zero if the two variables are uncorrelated, in which case - due to their normality - they also are independent.

In the case that one of the variables is vector-valued, say $\boldsymbol{y}$, one must use the total correlation

$$\rho_{xy} = \frac{\mathbb{C}\mathrm{ov}(\underline{x},\underline{\boldsymbol{y}})\mathbb{C}\mathrm{ov}(\underline{\boldsymbol{y}},\underline{\boldsymbol{y}})^{-1}\mathbb{C}\mathrm{ov}(\underline{\boldsymbol{y}},\underline{x})}{\sigma_x^2} \tag{1.39}$$

Mutual differential self-information has been used by Vosselman (1992) for structural matching in order to overcome the scale problems encountered by Boyer and Kak (1988) who used conditional differential self-information.

Fig. 1.4 provides an intuitive example, where the mutual information is large (in subfigures 2,4 and 5), caused by functional relationships, but the correlation coefficient cannot capture this relation, since it only reflects the degree of linear dependency.

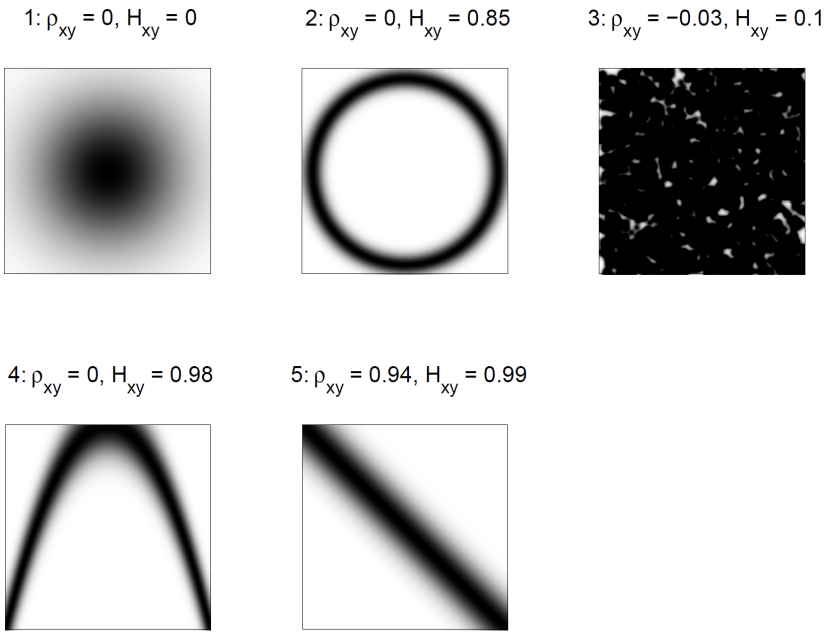


Figure 1.4: Correlation coefficient ρ_{xy} and mutual entropy H_{xy} for different 2D distributions $\mathrm{p}(x, y)$. Distributions 1 to 4 have (practically) zero correlation coefficient. Only the Gaussian and the nearly uniform distribution have mutual entropy close to zero, whereas the other blurry line-distributions reveal large mutual entropy

1.6 Principle of Maximum Entropy

In various situations we would like to know the distribution of a random variable, but only have partial knowledge, e.g., the range in which the random variable exists or some moments, e.g., the mean or the variance. Then it appears plausible to choose a distribution which fulfils these constraints, but does not add any information, i.e., we search for that distribution which, apart from the given constraints, maximizes the surprize. This leads to the concept of maximum entropy distributions. We show the derivation for an example and provide two important cases. Further cases are to be found in the exercises.

Example 1.6.10: ***Maximum entropy distribution in an interval.*** Assume the distribution $\mathrm{p}(x)$of a random variable is restricted to the $[a, b]$. Thus we have the condition

$$g(p, x) = \int_{x=a}^{b} \mathrm{p}(x)\,\mathrm{d}x - 1 = 0\,. \tag{1.40}$$

To find the distribution $\mathrm{p}(x)$ we maximize its entropy

$$\mathrm{H}(x; p) = -\int_{x=a}^{b} \mathrm{p}(x) \log \mathrm{p}(x)\,\mathrm{d}x\,. \tag{1.41}$$

under this constraint. For finding the maximum entropy distribution $\mathrm{p}(x)$, with the Lagrangian multiplier λ, we need to find the maximum of

$$\Phi(\mathrm{p}(x), \lambda) = -\int_{x=a}^{b} \mathrm{p}(x) \log \mathrm{p}(x)\,\mathrm{d}x + \lambda \left(\int_{x=a}^{b} \mathrm{p}(x)\,\mathrm{d}x - 1 \right) \tag{1.42}$$

w.r.t. to $\mathrm{p}(x)$. Using $1 = \int_{x=a}^{b} \frac{1}{b-a}\mathrm{d}x$ this function can be written as a function of p

$$L[p] := \int_{x=a}^{b} F(x, p)\mathrm{d}x \quad \text{with} \quad F(x, p) = -\mathrm{p}(x) \log \mathrm{p}(x) + \lambda \left(p - \frac{1}{b-a} \right)\,. \tag{1.43}$$

Using the results of calculus of variations (Weisstein, E. W., 2022) one can show that the function $p := \mathrm{p}(x)$ that minimizes $L[p]$ needs to satisfy $\mathrm{d}F/\mathrm{d}p = 0$. Using $\partial(y \log y)/\partial y = -\log y - 1$ we

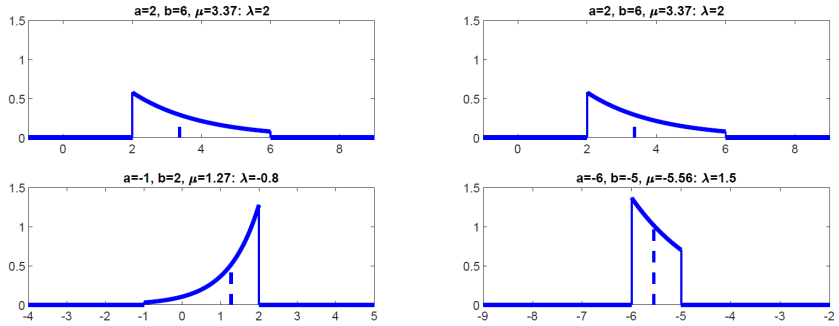


Figure 1.5: Maximum entropy distributions for given interval $[a, b]$ and mean μ. Observe the signs of the parameters. The parameter λ is determined numerically

explicitly obtain a constraint for p, namely

$$\frac{\mathrm{d}F}{\mathrm{d}p} = -\log \mathrm{p}(x) - 1 + \lambda = 0\,, \tag{1.44}$$

which results in

$$\mathrm{p}(x) = \mathrm{e}^{-1+\lambda}\,, \tag{1.45}$$

which is a constant. Thus we have, due to (1.40)

$$\mathrm{p}(x) = \frac{1}{b-a}\,. \tag{1.46}$$

◇

Hence, we have the following theorem:

Theorem 1.6.2: ***Maximum entropy distribution in a finite interval.*** If the random variable $\underline{x} \sim \mathrm{p}(x)$ is positive only in the interval $[a, b]$ then the maximum entropy distribution is a uniform distribution $\underline{x} \sim \mathcal{U}(a, b)$.

In a similar manner we find the following theorems, see the exercises.

Theorem 1.6.3: ***Maximum entropy distribution for positive values with given mean.*** If the random variable $\underline{x} \sim \mathrm{p}(x)$ is positive only in the interval $[0\infty)$ and has mean $\mathbb{E}(\underline{x}) = \mu$, then the maximum entropy distribution is an exponential distribution $\underline{x} \sim \mathrm{Exp}(\mu) = \mathrm{e}^{-x/\mu}/\mu, x \geq 0$. Exercise 1.5

Theorem 1.6.4: ***Maximum entropy distribution given mean and variance.*** If the random vector $\underline{\boldsymbol{x}} \sim p(\boldsymbol{x})$ has mean $\mathbb{E}(\underline{\boldsymbol{x}}) = \boldsymbol{\mu}_x$ and covariance matrix $\mathbb{D}(\underline{\boldsymbol{x}}) = \Sigma_{xx}$, then the maximum entropy distribution is a Gaussian distribution $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$. Exercise 1.6

The constraints can be combined, e.g., we also have the following result: Exercise 1.7

Theorem 1.6.5: ***Maximum entropy distribution with given range and given mean.*** If the random variable $\underline{x} \sim \mathrm{p}(x)$ is non-negative only in the interval $[a, b] \in \mathbb{R}^2$ and has mean $\mathbb{E}(\underline{x}) = \mu \in (a, b)$, then the maximum entropy distribution is an truncated exponential distribution

$$\mathrm{p}_{\mathrm{texp}}(x; a, b, \mu) = \frac{\mathrm{e}^{-\frac{x}{\lambda}}}{\lambda\left(\mathrm{e}^{-\frac{a}{\lambda}} - \mathrm{e}^{-\frac{b}{\lambda}}\right)} \quad \text{with} \quad x \in [a, b] \quad \text{and} \quad p_{\mathrm{texp}}(x) = 0 \text{ else} \tag{1.47}$$

with $\lambda = \lambda(\mu)$. The given parameters (a, b, μ) need not be positive.

Examples are given in the following figure. Given the parameters (a, b, μ) densities appear plausible.

1.7 Connection to Model Selection and Robust Estimation

An important application of information theory is model selection. We have seen that interpreting images (and data in general) amounts to fitting models to noisy data. In that

context, a natural question is what is a good model to represent a certain data set, or at least whether a specific model is a suitable representation. This question includes robust estimation, if one sees a robust estimator as one that fits a model consisting of two parts, one for the inliers and one for the outliers.

Information theory is a powerful tool to address these questions. Here, we chose to illustrate the principle with the simple example shown in Fig. 1.1. In an image of size $R \times R = 256 \times 256$ pixels, the coordinates of $n = 18$ points have been measured and provided with a resolution of $\varepsilon = \frac{1}{8}$ pixel. The question to be answered is: Are the points randomly distributed in the square or are some of them lying on a straight line. The four models shall be compared:

- models $\{A, B, D\}$: a straight line with parameters (a, b) that covers $n_I \in \{8, 12, 6\}$ inliers with a standard deviation of $\sigma \in \{5.8, 1.6, 8.7\}$ pixels, plus $n_I = n - n_O$ outliers uniformly distributed in the image
- model C: no line, only $n = n_O = 18$ random points uniformly distributed in the image

To code a uniformly distributed value in the interval R to a precision of ε, one requires $\mathrm{lb}\,(R/\varepsilon)$ bits. We thus get the following code lengths: for model C, we must code the two coordinates of all n image points, leading to a code length of

$$\Phi_C = 2n \cdot \mathrm{lb}\,\frac{R}{\varepsilon} = 2n(\mathrm{lb}\,R - \mathrm{lb}\,\varepsilon) = 36 \cdot (8+3) = 396\,[\mathrm{bits}]\,. \tag{1.48}$$

Observe, we obviously used (1.25) for the uniform distribution $\underline{x} \sim \mathcal{U}(0, R)$ with $p_x(x) = 1/R$ in the form

$$\mathrm{I}(\underline{x} \in [x, x+\varepsilon]) = -\mathrm{lb}\,\mathrm{p}(x) - \mathrm{lb}\,\varepsilon \tag{1.49}$$

here with $-\mathrm{lb}\,\mathrm{p}(x) = \mathrm{lb}\,R$. For the other three models A, B, and D, we can assume an optimal code, which requires $H(n_I/n)$ [bits] per point to distinguish inliers from outliers, see (1.14). Only the latter must be coded with their two coordinates as in (1.48), whereas for inliers we can instead code the line parameters (a, b), a single coordinate, say in x-direction, and the residual, say in y-direction. The parameters need only be coded with respect to their standard deviation $\sigma_p = \sigma/\sqrt{n}$, not necessarily to ε.

$$\begin{aligned}
\Phi_A &= \Phi_{par} + \Phi_{idx} + \Phi_O + \Phi_I \\
&= 2\mathrm{lb}\,\frac{R}{\sigma_p} + n\,\mathrm{H}\left(\frac{n_I}{N}\right) + 2n_O\mathrm{lb}\,\frac{R}{\varepsilon} + \left(\sum_{i=1,\dots,n_I}\left\{\frac{\mathrm{lb}\,\mathrm{e}}{2}\left(\frac{v_i}{\sigma}\right)^2 + \frac{1}{2}\mathrm{lb}\,2\pi\left(\frac{\sigma}{\epsilon}\right)^2 + \mathrm{lb}\,\frac{R}{\varepsilon}\right\}\right) \\
&= 2\mathrm{lb}\,\frac{R}{\sigma_p} + n\,\mathrm{H}\left(\frac{n_I}{N}\right) + 2n_O\mathrm{lb}\,\frac{R}{\varepsilon} + \left(\frac{\mathrm{lb}\,\mathrm{e}}{2}(n_I - 2) + n_I\left(\frac{1}{2}\mathrm{lb}\,2\pi\left(\frac{\sigma}{\epsilon}\right)^2 + \mathrm{lb}\,\frac{R}{\varepsilon}\right)\right) \\
&= 13.9 + 17.8 + 176 + (5.77 + 10 \cdot (3.9 + 8 + 6)) \approx 392\,[\mathrm{bits}]
\end{aligned} \tag{1.50}$$

using the relation $\mathbb{E}(\sum_i |\underline{\boldsymbol{v}}_i|^2/\sigma^2) = n_I - 2$ with the residuals $v_i := v_i(a, b)$ depending on the line parameters (a, b). The assumed resolution ε thus has no influence on the decision, since the resolution affects all coordinates, independent on whether the points are in- or outliers.

Model A has a shorter code and should be preferred. However, model D has the shortest description length. As the four main contribution (the bits for the outliers, the x-coordinates of the inliers, the model definition and the resulting y-coordinates), shown in Fig. 1.6 demonstrate, the B needs to many bits for the 12 outliers which cannot be compensated by the much less bits for the y-coordinates of the only 6 inliers.

The example demonstrates the usefulness of comparing models based on the coding length. The difference in bits directly can be interpreted as ratio of probabilities, that the different models and their realisations occur: E.g. configuration is $2^{396-390.2} \approx 56$ times more likely than the random configuration C.

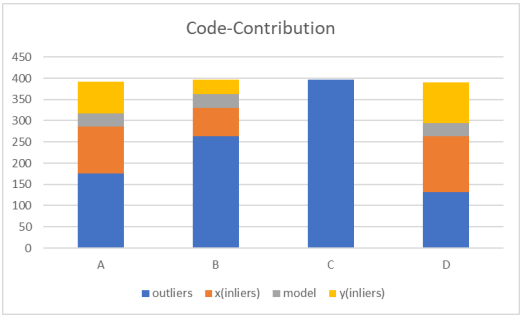


Figure 1.6: Contribution of outliers, inliers and model description on total description length. The differences of the total description lengths relatively small,

model	n	n_I	n_O	σ	Φ	$\Phi - \Phi_0$
A	18	10	8	5.8	392.18	-3.82
B	18	6	12	1.6	396.18	0.18
C	18	0	18	0.0	396.00	0.00
D	18	12	6	8.7	**390.21**	-5.78

Table 1.1: Modelselection, see Fig. 1.1: While assuming just 18 random points (case C) leads to a description length of 396 [bits], the model D is suggested as the optimal model, having a wide band with $\sigma = 8.7$, with a description length of appr. 383 [bits]. Models A and B assumes 10 and 6 inliers, respectively, lying on partly much narrower bands (with $\sigma = 5.8$ and $\sigma = 1.6$), thus with 8 and 12 outliers, respectively. Only model A and D have a shorter description length than assuming random points, see Fig. 1.6. The description length of model B is only slightly larger, than when assuming randomly distributed points

Generally, instead of specifying probabilities for the possible outcomes of an experiment, we therefore also could use a coding scheme for generating each possible outcome, and use this as surrogate for a probabilistic model. This especially is useful for discrete and nested models, see (Kolbe, 2000).

An alternative interpretation of the *model selection* according to (1.50) leads to a *robust estimator* of the line parameters

model selection as robust estimation

$$(\hat{a}, \hat{b}) = \text{argmin}_{a,b} \sum_{i=1}^{n} \rho(v_i(a,b)) \tag{1.51}$$

since one can show, that Φ_A can be written as

Exercise 1.4

$$\Phi_A(a,b) = \lambda_1 + \lambda_2 \sum_{i=1}^{n} \rho(v_i(a,b)) \quad \text{with} \quad \rho(v_i) = \min(v_i^2(a,b)/\sigma^2, k^2) \tag{1.52}$$

and a suitable choice of the three parameters $\lambda_i, i = 1, 2$ and k. Hence, minimising the code length over all possible lines $ax + by = 1$ with uncertainty σ is equivalent to fitting the line parameters with the *truncated least squares* estimator.

1.8 Exercises

1. Assume two binary variables $\underline{A}$ and $\underline{B}$ have the joint probabilities $\mathrm{P}(A, B)$ given in the following table:
   (a) Determine the mutual self-information $\mathrm{I}(A = 1, B = 0)$ of the two messages $\underline{A} = 1$ and $\underline{B} = 0$. Explain the result.
   (b) Determine the mutual entropy/mutual information $\mathrm{H}(\underline{A}; \underline{B})$. Explain, why $\mathrm{H}(\underline{A}; \underline{B})$ can be positive in spite of $\mathrm{I}(A = 1, B = 0)$ being negative.

$P(A,B)$	$A=0$	$A=1$
$B=0$	0.50	0.05
$B=1$	0.15	0.30

Table 1.2: Joint probabilities

2. Take the file `MIT.png`, convert the image into a gray-level image, store this image and use its histogram. Determine the expected number N of bits you need to code the image. Compare it to the length of the image file. Is the size of the stored image smaller than N? Why? Discuss.
   Remark: The entropy is $\mathrm{H} = 5.1351$, the number of bits required for coding is $N = 256^2\mathrm{H} = 336533$. The file has a size of 37.6 Kb, thus requires approximately 11% of the number N of bits. The reason is, that the mutual dependencies of the grey-values are not taken into account. ◇

3. Refer to the definition (1.19) of the the Kullback-Leibler divergence, and show that it does not fulfil the triangle inequality. As an example, take three binary random variables with $\mathrm{P}(\underline{x}=1) = 0.1$, $\mathrm{Q}(\underline{x}=1) = 0.7$, and $\mathrm{R}(\underline{x}=1) = 0.8$. Why is a single example sufficient for the proof?

4. Prove (1.52) under the assumption that all parameters, except the residuals v_i are given. Give explicit expressions for λ_1, λ_2 and k.

5. Show that the exponential distribution $\mathrm{Exp}(\mu)$ is the minimum entropy distribution for a non-negative random variable with mean μ. *Hint:* Bring the optimization function Φ, corresponding to (1.42), into the form $L[p] = \int_{x\geq 0} F(x,p)\mathrm{d}x$.

6. Prove Theorem 1.6 in the following steps.
   (a) Show that the Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ is the minimum entropy distribution for a random variable with mean μ and variance σ^2. *Hint:* Bring the optimization function Φ, corresponding to (1.42), into the form $L[p] = \int_x F(x,p)\mathrm{d}x$.
   (b) Rotate the coordinate system, such that the n-vector $\underline{\boldsymbol{y}} = \boldsymbol{R}\underline{\boldsymbol{x}}$ has a diagonal covariance matrix $\boldsymbol{\Sigma}_{yy}$ and use the result of item 6a. Derive the differential entropy $\mathrm{H}(\underline{\boldsymbol{y}})$. Express it as a function of the variances $\sigma^2_{y_i}, i = 1,...,n$.
   (c) Use the result of 6b to prove that the entropy of $\underline{\boldsymbol{x}}$ is

$$\mathrm{H}(\underline{\boldsymbol{x}}) = \frac{1}{2}\log|\boldsymbol{\Sigma}_{xx}| + \frac{n}{2}(1+\log(2\pi))\,. \tag{1.53}$$

   Use this result to prove Theorem 1.6.

7. Refer to Theorem 1.6 and show that
   - for given interval $[a,b], b > a$ and mean μ the maximum entropy distribution is a truncated exponential.
   - Show, that it can be written in the form of Eq. (1.47).
   - Show that the mean of this truncated distribution is

$$\mu_{\mathrm{texp}}(a,b,\lambda) = \lambda + \frac{a\,\mathrm{e}^{-\frac{a}{\lambda}} - b\,\mathrm{e}^{-\frac{b}{\lambda}}}{\mathrm{e}^{-\frac{a}{\lambda}} - \mathrm{e}^{-\frac{b}{\lambda}}}\,. \tag{1.54}$$

   Observe, that solving (1.54) cannot (easily) be solved for λ algebraically.

2 Signal Theory

Signal Theory is the essential tool for understanding linear filters of one- or multi-dimensional signals. These lecture notes use the relation of cyclical matrices and their eigenvalue decomposition as basis for explaining the Fourier transformation. The lectures cover correlation and convolution, the Fourier transformation for discrete and continuous, and cyclical and infinite one- and multi-dimensional signals, the sampling theorem and other image transforms.

2.1 Preface

Lectures on Signal theory provided a basis for understanding linear filters applied to time series and digital images. Together with Konrad Schindler, we augmented the lecture notes for a book section. The motivation for using signal theory goes back to my work on variance component estimation for observed autoregressive processes Förstner (1985), which exploits the relation of cyclical matrices and their eigenvalue decomposition, essentially being the discrete Fourier transform of periodic signals. This appeared to be a good starting point for a lecture on Signal Theory.

Since signals often are non-periodic and finite, we need to specify the basic operations convopution and correlation more in detail, see Sect.92 in the note on convolutional networks.

2.2 Introduction

Images can be interpreted as one or more dimensional *signals* in two or more dimensions. *signal*
Examples are black-and-white or depth images, colour images, multispectral images, tomographic images, or colour video sequences, see Fig. 2.1. The table gives examples for

Figure 2.1: Digital images. From left to right: Black and white image $g(x, y) : \mathbb{R}^2 \to \mathbb{R}$, color image $\boldsymbol{g}(x, y) : R^2 \to R^3$, some images from a video sequence $\boldsymbol{g}(x, y, t) : R^3 \to R^3$

multi-dimensional signals $\boldsymbol{y}(\boldsymbol{x})$, we might encounter. This representation enforces a certain (freely chosen) view onto the signal, since the same data, e.g., a grey value video, may be represented as a scalar depending on space and time, $g(\boldsymbol{x}, t)$ or as a matrix $\mathsf{G}(t)$ changing over time. Mathematically, we could write $g(\boldsymbol{x}, t)$ as $g(\boldsymbol{x})(t)$ or as $g(t)(\boldsymbol{x})$ and group the arguments as needed, e.g., as a spatially varying time-vector, namely $\boldsymbol{g}(\boldsymbol{x})$, where the vector $\boldsymbol{g}(\boldsymbol{x}_0)$ represents the time signal $\{g(t)\}(\boldsymbol{x}_0)$ at position x_0. In all cases the data can be represented as higher dimensional matrices, i.e., generally tensors, being scalars $y(x_1, ..., x_D)$ as a function of several variables, with the classical special cases vectors $D = 1$ and matrices $D = 2$. The dimension D also is called the "rank" of the tensor T.[1] Specifically digital images can be seen as discrete two dimensional signals which are

D	example
1	change of temperature over time $T(t)$, a line in a grey value image $g(i)$
2	a digital elevation model $H(X, Y)$, a grey value image $g(i, j)$ change of data-vector over time $\boldsymbol{d}(t)$, a line in a color image $\boldsymbol{g}(i, \cdot)$
3	a color image $\boldsymbol{g}(i, j)$, a multi spectral image, a magnet resonance image (MRI) $R(i, j, k)$, a grey value video $g(i, j, t) = \mathsf{G}(t)$
4	a color video $\boldsymbol{g}(i, j, t)$, MRI image sequence
5	a rectangular set $\boldsymbol{g}_{k,l}(i, j)$ of color images

Table 2.1: Image and image sequences as D-dimensional signals

derived from an underlying continuous image by discretization, see Fig. 2.1.

Many useful operations on signals are linear, such as contrast enhancement or noise suppression. Linear signal theory allows to analyse and interpret such operations.

The relevance of linear systems theory is, that optical systems can be modelled as linear shift-invariant systems, see Fig. 2.3.

The blurred image c in Fig. 2.3 can be interpreted as being a transformed version of the ideal image a. If the image plane is shifted w.r.t. to the optics both functions, b and c, are shifted by the same amount. Moreover, of the image c will have double the values if the values of b are doubled.

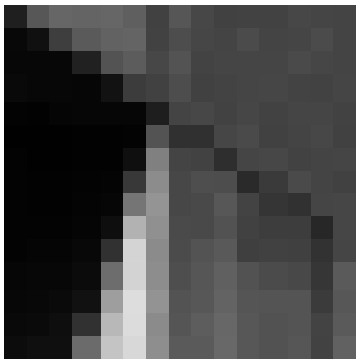
Formally, linear shift-invariant operators $\mathcal{L}_a$ are characterized in the following manner: let the system, which generates $c(x, y)$ from $b(x, y)$ be the operator $\mathcal{L}_a$, hence

$$c(x, y) = \mathcal{L}_a(b(x, y)) \,. \tag{2.1}$$

Then $\mathcal{L}_a$ is *linear* and *shift-invariant*, if

$$\mathcal{L}_a(\alpha_1 b_1 + \alpha_2 b_2) = \alpha_1 c_1 + \alpha_2 c_2 \tag{2.2}$$

[1] We do not distinguish tensors of rank > 2 in the notation. We do treat complex numbers as scalars.



31	87	102	100	102	95	65	86	71	65	66	66	71	69	68
8	16	62	90	98	100	66	77	70	68	75	68	69	75	68
7	7	7	33	83	92	71	83	67	69	69	69	74	70	68
9	7	7	6	18	60	65	77	65	67	69	70	67	66	69
3	2	5	6	6	6	30	68	71	66	70	66	69	69	66
1	1	1	2	2	3	83	48	48	70	74	65	68	71	65
3	1	1	2	2	15	128	68	66	46	69	71	66	68	69
3	2	2	3	4	56	140	71	78	77	42	58	74	68	65
3	2	2	4	6	117	148	71	73	87	67	53	49	69	70
3	3	3	5	14	178	144	74	73	82	65	61	61	39	70
3	5	5	8	49	212	139	75	83	89	63	65	64	50	69
6	7	7	11	109	211	141	78	86	96	82	75	72	54	90
7	8	10	17	158	221	147	83	87	99	85	84	84	59	87
9	10	14	49	184	220	139	85	93	103	87	82	84	67	90
12	15	15	107	196	216	142	90	90	104	87	82	86	66	97

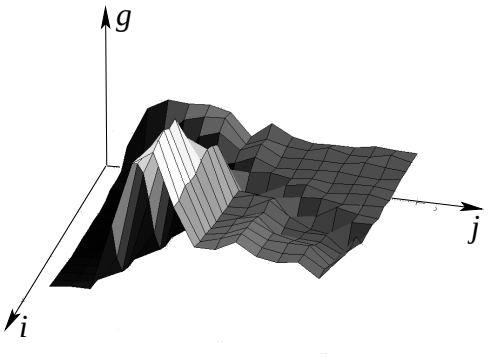


Figure 2.2: Image as signal. **Upper left:** Original grey level image. **Upper right:** Blow up of subsection. **Lower left:** Grey values of subsection. **Lower right:** Grey values as digital surface model

and

$$\mathcal{L}_a(b(x-u, y-v)) = c(x-u, y-v) \tag{2.3}$$

holds.

Signals in signal theory are scalar or complex functions of one or several variables. Systems for operating on signals are graphically represented as in Fig. 2.4. Signal theory has evolved for analysing of time signals $x(t)$, especially electrical and acoustical signals, clearly documented in (Shannon and Weaver, 1949), (?), and triggered by the development of television was generalized to two dimensional spatial signals $g(x, y)$. Specific conditions, which needed to be fulfilled for time signals, e.g., causality, i.e., that a reaction cannot start before its cause, are not relevant for two-dimensional signals. A special group of operation on signals are linear shift-invariant operators often simply called *filters*. This results from the fact that linear shift-invariant operators filter certain parts of the signal while letting other pass. Such filters can be analysed completely and elegantly using linear systems theory.

An essential tool is their spectral analysis: for time signals this results in an additive decomposition in cos- and sin-waves with different frequencies, see Fig. 2.5.[2] Hence, we have the representation, e.g., for $b(t)$

$$b(t) = \sum_{n=0}^{\infty} B_n \sin(2\pi\, n\, t + \varphi_n) \tag{2.4}$$

Often a representation with complex numbers is of advantage:

$$b(t) = \sum_{n=-\infty}^{\infty} B'_n \, \mathrm{e}^{j\, 2\pi\, n\, t} \tag{2.5}$$

[2] For spatial signals the spatial frequencies are multidimensional.

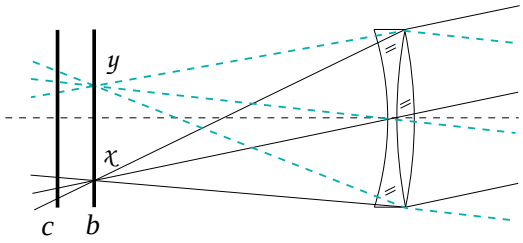


Figure 2.3: Optical realization of a linear shift-invariant system: Given the image function b, then shifting the image plane parallel or perpendicular to the optical axes leads to an image function c, which linearly depends on the function b. The effect of the imaging system is the same at all points of the image, specifically at the two positions x and y

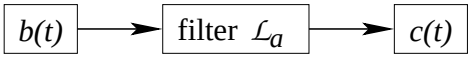


Figure 2.4: Block diagram for a filter. The name or the type of the filter operation is shown in the box.

where, in order to obtain real valued signals, the coefficients need to be complex numbers, thus, contain amplitude and phase. Exercise 2.1

The essential property of linear filters is: Linear filters only change amplitudes and phases. This allows to easily interpret their effect, if we analyse the signals and the filters in their frequency domain. For example: Smoothing a signal only reduces the amplitudes of the waves, and leaves the phases unchanged.

We distinguish four types of signals: signals may be continuous or discrete, and they may have an infinite or finite domain. Since the boundaries of a signal with finite domain cannot be handled in a conceptually clear way,[3] we assume the signal to be periodic, i.e., given signal, e.g., $f(x), x \in [0, a]$ is one period of an infinite periodic signal, e.g., $f(x) = f(x + na), n = 0, \pm 1, \pm 2, ...$

Hence we have the following four types of signals, making them explicit for one and two arguments.

1. Infinite continuous one or multi-dimensional functions or images

$$f(x) \quad \text{with} \quad x \in \mathbb{R} \quad \text{or} \quad f(x, y) = f(\boldsymbol{x}), \quad \text{with} \quad \boldsymbol{x} \in \mathbb{R}^2 \tag{2.6}$$

with indices from the end of the alphabet, which can be generalized to $\boldsymbol{x} \in \mathbb{R}^D$. [4]

[3] We will discuss several ways how to handle the boundary of signals with finite domain in Sect. XXX.

[4] The dimension of the entity is to be taken from the context. We reserve capital letters, e.g., the function G in times font for the Fourier transform of the signals, e.g., of g.

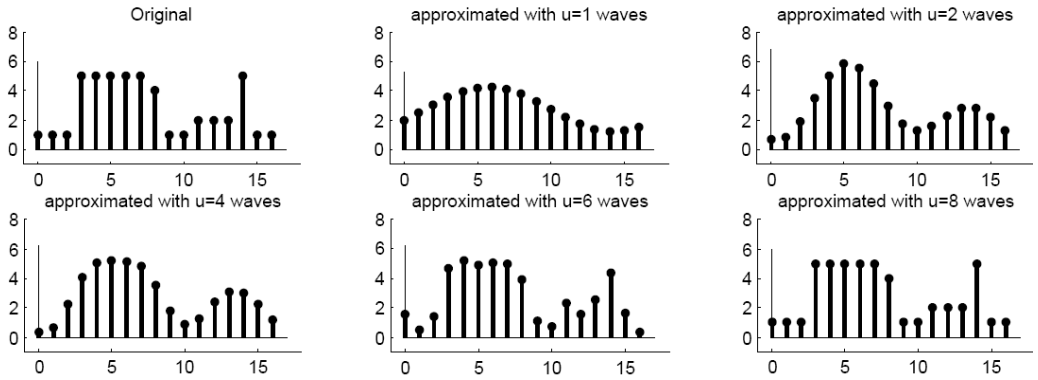


Figure 2.5: Approximation of a signal by periodic functions. Upper left: Original signal. Following sub images: approximation with $U = 1, 2, 4, 6, 8$ waves

2. Periodic continuous functions or images

$$f(x) = f(x+u) \quad \text{with} \quad x \in [0, u]\,, \tag{2.7}$$

or in two dimensions

$$f(\boldsymbol{x}) = f(\boldsymbol{x}+\boldsymbol{u}) \quad \text{with} \quad \boldsymbol{x} \in [0, u_1] \times [0, u_2]\,. \tag{2.8}$$

3. Infinite discrete one or two-dimensional functions or images

$$f(i) \quad \text{with} \quad i \in \mathbb{Z} \quad \text{or} \quad f(i,j) = f(\boldsymbol{k})\,, \quad \text{with} \quad \boldsymbol{k} = [i,j]^{\mathsf{T}} \in \mathbb{Z}^2 \tag{2.9}$$

with indices from the middle of the alphabet. Collecting the discrete entries in vectors or matrices results in representations $\boldsymbol{f} = [f_i] = [f(i)]$ or, in two dimensions, $\boldsymbol{f} = [f_{ij}] = [f(i,j)]$. Again, generalization to higher dimensions can be achieved by $\boldsymbol{k} \in \mathbb{Z}^d$.

4. Periodic discrete functions or images

$$f(i) = f(i+n) \quad \text{with} \quad i \in [0, n-1]\,, \tag{2.10}$$

or in two dimensions

$$f(\boldsymbol{k}) = f(\boldsymbol{k}+\boldsymbol{m}) \quad \text{with} \quad \boldsymbol{k} \in \{0, m_1 - 1\} \times [\{0, m_2 - 1\}\,. \tag{2.11}$$

The notation is chosen such that continuous and discrete signals can be distinguished from their argument, e.g., x or i, whereas no notational distinction is made between periodic and nonperiodic signals.

This chapter discusses important aspects of linear systems theory. We will start with discrete periodic one-dimensional signals. Linear operators then are discrete cyclical convolutions which easily can be represented as multiplications with *cyclical matrices*. Since the eigenvectors of cyclical matrices are periodic functions with different spatial frequencies, we obtain a decomposition of the signal as a weighted sum of periodic basis functions. This leads to what is called the discrete Fourier transformation (DFT). Applying the analysis to stochastic signals gives us insight into the correlation structure of such signals.

We then generalize the signal analysis for periodic discrete signals to the other types of signals (continuous, non-periodic), finally leading to the Fourier transformation[5] of infinite non-periodic continuous signals. We use it for the analysis of the properties of basic filters, of the sampling process, and for the derivation of the sampling theorem.

The basis functions of the Fourier transformation are non-zero over the complete domain, except for a countable set of points. Hence, each value of the Fourier transform depends on all values of the signal. This does not allow us to analyse the spectral properties of the signal locally. Therefore, we finally discuss signal representations with basis function with finite support, what are called *wavelets*. They have the advantage of characterizing the spectral properties of the signal locally, at the same time can be determined as efficient as the Fourier transformation.

Remark: For complex numbers $z = x + \mathrm{i}y$ with $\mathrm{i} = \sqrt{-1}$ we have: real part $\Re(x + \mathrm{i}y) = x$, imaginary part $\Im(x + \mathrm{i}y) = y$, product $(a + \mathrm{i}b)(c + \mathrm{i}d) = (ac - bd) + \mathrm{i}(bc + ad)$, conjugate complex number $z^* = (x + \mathrm{i}y)^* = x - \mathrm{i}y$, absolute value $|z| = |x + \mathrm{i}y| = \sqrt{x^2 + y^2} = \sqrt{z\,z^*}$, exponential function $\mathrm{e}^{i\phi} = \cos\phi + \mathrm{i}\sin\phi$ and hence the relation between Cartesian and polar coordinates $|z| = x + \mathrm{i}y = |z|\mathrm{e}^{\mathrm{i}\phi} = |z|(\cos\phi + \mathrm{i}\sin\phi)$ with $r = \sqrt{x^2 + y^2}$ and $\phi = \operatorname{atan2}(x, y)$. If we have a matrix $\boldsymbol{A} = [a_{ij}]$ with complex entries the matrix $\boldsymbol{A}^* = [a^*_{ji}]$ is the transpose matrix with conjugate complex elements. Often we use $\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$. ◇

[5] The notion Fourier transformation in mathematics is used in a more general sense, namely when representing a function as a sum of orthogonal basis functions, not necessarily periodic ones, as e.g., using orthogonal polynomials.

2.3 Convolution and Correlation

One of the basic tasks in signal processing is template matching, namely to locate the position of a specific signal, called the template, say $a(x)$, in a longer signal, say $b(x)$. Let us for simplicity assume, the domain of $a(x)$ is limited to a small range $[-1, +1]$ around 0, i.e., we assume do not know the values of $a(x)$ outside this interval, see Fig. 2.6. Then it

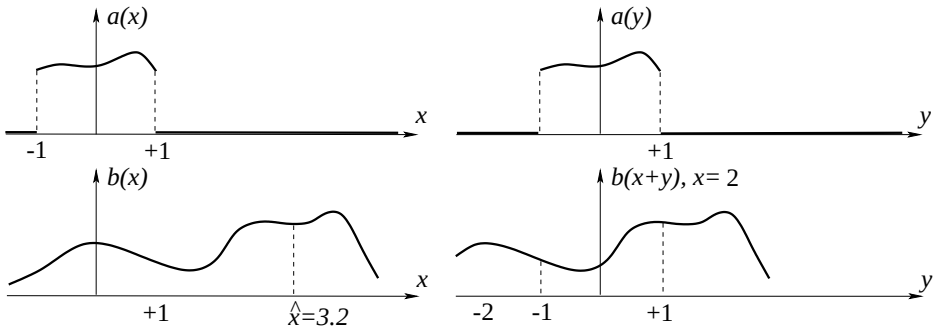


Figure 2.6: Principle of template matching: **Left:** Given the template $a(x)$, assuming no information about the values outside the interval $[-1, +1]$ and the signal $b(x)$. **Right:** Finding the best location of the template in $b(x)$ can be achieved by determining the maximum inner product of the template $a(y)$, now assuming it to have value 0 outside the interval $[-1, +1]$ with a shifted version of $b(x + y)$, the variable change used to express the inner product.

is intuitive to define a similarity measure $c(x)$ between the template a and all subsections $[x-1, x+1]$ of length 2 of the signal b and take the optimum of the this similarity measure $c(x)$ as most likely position.

Here, we adopt the inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, which measures the similarity of the two vectors and is $\langle a, b\rangle = \boldsymbol{a}^\top \boldsymbol{b} = \sum_i a_i b_i = |a|\,|b|\,\cos(\boldsymbol{a}, \boldsymbol{b})$ which is large, in case the two vectors differ slightly, since then the angle between the vectors is small. This inner product can be generalized to functions $a(x)$ and $b(x)$, by replacing the sum by the integral of the products. Hence, formally we need to determine

$$c(x) = \langle a(y), b(x + y)\rangle = \int_{y=x-1}^{x+1} a(y)b(x + y)\mathrm{d}y \tag{2.12}$$

This can be written as an infinite integral if we assume $a(x) = 0$ outside the interval $[-1, 1]$. The result is the "correlation" of the template $a(x)$ with the signal $b(x)$,

$$c(x) = a(x) \circledast b(x) := \int_{y=-\infty}^{\infty} a(y)b(x + y)\mathrm{d}y\,. \tag{2.13}$$

Though this correlation operation is linear and shift invariant, is has the disadvantage, that it is not commutative. As we will see in the next section, this disadvantage easily can be eliminated, by first mirroring the template and then applying correlation, thus determining $\int_{y=-\infty}^{\infty} a(-y)b(x + y)\mathrm{d}y$ or by defining an operation, called "convolution".

Remark: The notion "correlation" in signal theory means the inner product of two mutually shifted signals, and is not to be confused with the normalized correlation coefficient used in statistics, though both concepts are closely related: the estimated correlation coefficient of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is $\rho(\bar{\boldsymbol{a}}, \bar{\boldsymbol{b}}) = \langle \bar{\boldsymbol{a}}, \bar{\boldsymbol{b}}\rangle / (|\bar{\boldsymbol{a}}|\,|\bar{\boldsymbol{b}}|)$, where the $\bar{\boldsymbol{a}}$ contains the elements of $\boldsymbol{a}$ reduced by the mean of the values a_i. ◇

2.3.1 Definition of Convolution

A fundamental theorem of linear systems theory is the following, expressed for infinite continuous signals. It allows to characterize all linear shift invariant operators, especially filters we regularly apply for manipulating images.

Theorem 2.3.6: ***Linearity and shift-invariance operators.*** If an operator $\mathcal{L}_a$ operating on $b(x)$, leading to $c(x) = \mathcal{L}_a(b(x))$ is linear and shift-invariant it can be represented as a convolution

$$c(x) = \mathcal{L}_a(b(x)) = a(x) * b(x) = \int_{y=-\infty}^{\infty} a(y)b(x-y)\mathrm{d}y \tag{2.14}$$

with some adequate function $b(z)$. Sometimes we refer to the first function, here $a(x)$, as the convolution kernel.

As an example, taking the moving average

$$c(x) = \mathcal{L}_w(b(x)) = \frac{1}{w}\int_{x=-w/2}^{+w/2} b(x)\mathrm{d}x = \int_{y=-\infty}^{\infty} r_w(y)b(x-y)\mathrm{d}y \tag{2.15}$$

It can be written as the convolution of the rectangle function $r_w(x)$ with $b(x)$. It a linear filter, since the integration is a linear and shift-invariant operator. Exercise 2.3

Remark: The notion *convolution* is derived from the Latin word "convolvere", which means "to roll together" and describes the fact that one of the two functions, here $b(x)$, is mirrored, i.e., it is used as $b(x-y)$ with y the integration variable, see Fig. 2.7. In two dimensions we need

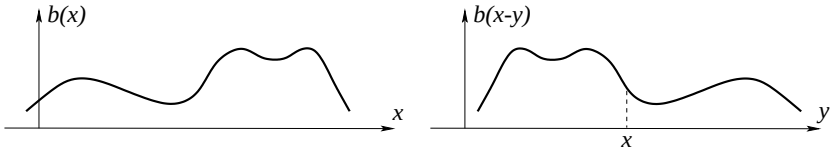


Figure 2.7: For convolution we mirror $b(x)$ at a specific x, to obtain $b(x-y)$ as a function of y, and take the inner product of the convolution kernel $a(y)$ and the mirrored function $b(x-y)$

to mirror at the origin $b(\boldsymbol{x}-\boldsymbol{y})$. If the convolution kernel is symmetric, hence $a(x) = a(-x)$, the effect is not visible, since then $a(x) * b(x) = \int_y a(-y)b(x+y)\mathrm{d}y = \int_y a(y)b(y+x)\mathrm{d}y$, and the result $c(x)$ is the scalar product of the function $b(y)$ shifted by x to the left with the convolution kernel $a(y)$. ◇

We have the following definitions of convolution for the other types of signals:

$$c(x) = a(x) * b(x) = \int_{y=0}^{a} a(y)b(\mathrm{mod}(x-y,u))\,\mathrm{d}y \tag{2.16}$$

$$c(i) = a(i) * b(i) = \sum_{i=-\infty}^{\infty} a(j)b(i-j) \tag{2.17}$$

$$c(i) = a(i) * b(i) = \sum_{i=0}^{n-1} a(j)b(\mathrm{mod}(i-j,n))\,. \tag{2.18}$$

We use the operator $*$ for indicating a convolution, overloading it for all types of signals. Often, we do not make the arguments explicit and simply write

$$c = a * b\,. \tag{2.19}$$

The modulo-function, namely $\mathrm{mod}\,(x,y) = x - |y|\lfloor x/|y|\rfloor \in [0,y]$, used in (2.16) and (2.18) enforce the argument of $b(\cdot)$ to lie in the intervals $[0,u)$ and $[0,n-1]$, respectively. These are the ranges of the function $b(\cdot)$; for periodic functions we have, e.g., $\mathrm{mod}(x+ku,u) = x$ for some $x \in [0,u)$.

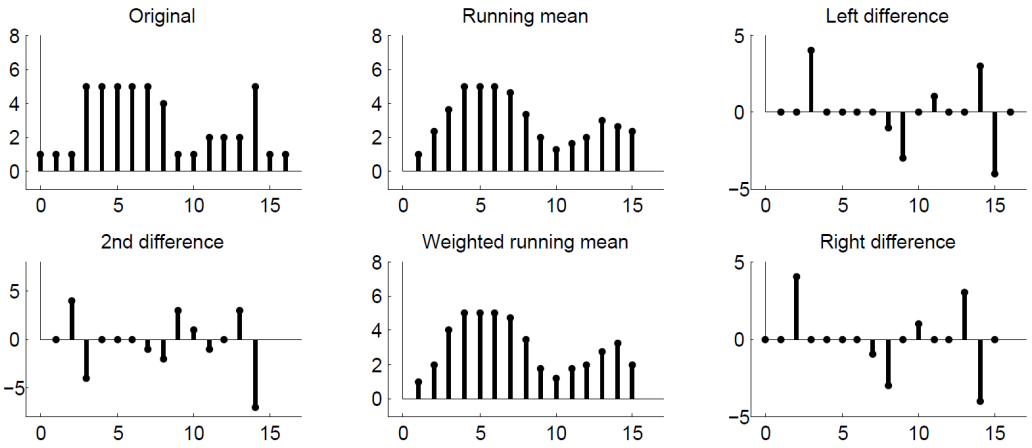


Figure 2.8: Linear filters. **Upper left**: Original x. **Upper mid**: running mean $R_3 * x$. **Lower mid**: weighted running mean $B_2 * x$. – Compare the slight but significant difference at $i = 14$. **Upper right**: Left difference $d_- * x$. **Lower right**: Right difference $d_+ * x$. — Observe the mutual shift. **Lower left**: 2nd difference $c * x$. Observe zeros and local (absolute) maximum at $i = 14$

Example 2.3.11: ***Linear filters used in image processing.*** We give a few examples for linear filters used in image processing. We show the results on a finite discrete signal, for which we assume the values outside the given range are unknown.

- **Moving average:**

$$y(i) = (x_{i-1} + x_i + x_{i+1})/3 \quad \text{or} \quad y = R_2 * x \quad \text{with} \quad R_3 = \frac{1}{3}[1\ \underline{1}\ 1]\ . \tag{2.20}$$

  Observe the mean of the three neighbouring values at $i = 14$: It is counter intuitive, that the mean at $i = 13$ is smaller than the mean at $i = 15$. Therefore, we have a look at a weighted moving average:

- **Weighted moving average.** here we weight the central value higher than the two neighbours:

$$y(i) = (x_{i-1} + 2x_i + x_{i+1})/4 \quad \text{or} \quad y = B_2 * x \quad \text{with} \quad B_2 = \frac{1}{4}[1\ \underline{2}\ 1]\ . \tag{2.21}$$

  The chosen weighting as a positive effect: The weighted mean at $i = 13$ is larger than the weighted mean at $i = 15$, which appears plausible.
  Remark: We will identify the reason for this behaviour of the two filters, when analysing their response to different wave lengths. ◇

- **Left and right difference:** Exercise 2.2

$$y(i) = x(i) - x(i-1) \quad \text{or} \quad y = d_- * x \quad \text{with} \quad d_- = [\underline{1}\ -1]\ . \tag{2.22}$$

  and

$$y(i) = x(i+1) - x(i) \quad \text{or} \quad y = d_+ * x \quad \text{with} \quad d_+ = [1\ \underline{-1}]\ . \tag{2.23}$$

  Both operations yield the gradient of a linear function, e.g., for $x(i) = a + bi$ we obtain $y(i) = b$.

- **Second difference**, which yields the second derivative for a quadratic function, which is an approximation for the curvature

$$z(i) = x(i-1) - 2x(i) + x(i+1) \quad \text{or} \quad z = c * x \quad \text{with} \quad c = [1\ -2\ 1]\ . \tag{2.24}$$

  E.g. for the quadratic function $x(i) = a + bi + ci^2$ we obtain $z(i) = 2c$.

- **Convolving two kernels** may be useful. E.g. if we convolve d_- with d_+ we obtain

$$c(i) = \sum_{j=-1}^{0} d_-(j)\, d_+(i-j) \quad \text{or} \quad c = d_- * d_+\,. \tag{2.25}$$

  This strongly motivates the analysis of filters w.r.t. their generating components, here the intuitive definition of the second differences as a convolution of two first differences.

◇

2.3.2 Properties of Convolution

Before analysing and applying convolutions, we want to collect the essential properties of this operator. We show the properties here for infinite discrete signals with vectors $\boldsymbol{a} = [a(i)]$ and $\boldsymbol{b} = [b(i)]$

1. **Commutativity**: the convolution is *commutative*:

$$\boldsymbol{a} * \boldsymbol{b} = \boldsymbol{b} * \boldsymbol{a}\,. \tag{2.26}$$

   This can be seen by replacing variables in (2.17), namely $j \Leftarrow (i-j)$ in $b(i-j)$, thus $(-j) \Leftarrow (j-i)$ and therefore $(i-j) \Leftarrow j$ in $a(j)$. This yields $b(i) * a(i)$, when following the definition in (2.17).
   Similarly, we can show the following:

2. **Associativity**

$$\boldsymbol{a} * \boldsymbol{b} * \boldsymbol{c} = (\boldsymbol{a} * \boldsymbol{b}) * \boldsymbol{c} = \boldsymbol{a} * (\boldsymbol{b} * \boldsymbol{c})\,. \tag{2.27}$$

3. **Distributivity**

$$(\boldsymbol{a} + \boldsymbol{b}) * \boldsymbol{c} = \boldsymbol{a} * \boldsymbol{c} + \boldsymbol{b} * \boldsymbol{c}\,. \tag{2.28}$$

4. **Multiplication with a scalar**

$$\lambda(\boldsymbol{a} * \boldsymbol{b}) = (\lambda \boldsymbol{a}) * \boldsymbol{b} = \boldsymbol{a} * (\lambda \boldsymbol{b}) \tag{2.29}$$

5. **Zero-index**: Infinite discrete signals have index in the range $(-\infty, ..., 0, ..., +\infty)$. If not clear from the context we indicate the index 0 by underlining:

$$\boldsymbol{a}^{\mathsf{T}} = [\ldots,\ a_{-2},\ a_{-1},\ \underline{a}_0,\ a_1,\ a_2,\ \ldots] \tag{2.30}$$

   If not stated otherwise, elements which are not given are assumed to be zero. As an example, take the one- and two-dimensional signals

$$\boldsymbol{w}^{\mathsf{T}} = \frac{1}{4}[\ldots\ 0\ 1\ \underline{2}\ 1\ 0\ \ldots] = \frac{1}{4}[1\ \underline{2}\ 1] \quad \text{or} \quad \boldsymbol{w} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & \underline{4} & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.31}$$

6. The one-element is the **unit impulse**

$$\boldsymbol{\delta} = [\ldots\ 0\ 0\ \underline{1}\ 0\ 0\ \ldots] \quad \text{or} \quad \boldsymbol{\delta} = \begin{bmatrix} \vdots\vdots & \vdots & \vdots\vdots \\ \ldots & \underline{1} & \ldots \\ \vdots\vdots & \vdots & \vdots\vdots \end{bmatrix}. \tag{2.32}$$

7. The **translation** of a function $f(i)$ by l or of a two dimensional function $f(i,j)$ by (l,m) can use the convolution with the unit impulse

$$f(i-l) = f(i) * \delta(i-l) \quad \text{or} \qquad f(i-l, j-m) = f(i,j) * \delta(i-l, j-m) \tag{2.33}$$

8. The **inversion** of convolution with $\boldsymbol{a}$ is possible if $\sum_i a(i) \neq 0$ and certain additional constraints are fulfilled.[6] The inverse element $a^{-1}(i)$ of $a(i)$ then can be used to solve $c(i) = a(i) * b(i)$ for $b(i)$ by *deconvolution*

$$b(i) = a^{-1}(i) * c(i) = c(i) * a^{-1}(i)\,. \tag{2.34}$$

[6] The Fourier spectrum then must be non-zero everywhere.

9. The **correlation**, as introduced in the motivation, differs from convolution by the sign of the first function. Hence, we have

$$c(x) = a(x) \circledast b(x) = a(-x) * b(x) = \int_{y=-\infty}^{\infty} a(y)b(x+y)\mathrm{d}y\,. \tag{2.35}$$

Since correlation is the basic tool for solving the template matching problem, convolution can be interpreted as correlation with the mirrored template, i.e., the mirrored first function:

$$c(x) = a(x) * b(x) = a(-x) \circledast b(x)\,. \tag{2.36}$$

It can easily be shown, that correlation is not commutative, but the other properties of convolution are still valid, such as associativity and distributivity.

If the first function $a(x)$ is symmetric, correlation and convolution are identical.

Convolution appears in many other contexts:

- The density function p_z of the sum $\underline{z} = \underline{x} + \underline{y}$ of two random variables $\underline{x}$ and $\underline{y}$ is the convolution of the two density functions p_x and p_y:

$$\underline{z} = \underline{x} + \underline{y} \qquad \Leftrightarrow \qquad p_z = p_x * p_y\,. \tag{2.37}$$

This holds for continuous and discrete random variables.

- The multiplication of two polynomials $p(x) = \sum_i^k a_i x^i$ and $q(x) = \sum_{i=0}^l b_i x^i$, is the polynomial $r(x) = p(x)q(x) = \sum_{i=0}^{k+l} c_i x^i$ with the coefficients

$$[\underline{c}_0, ..., c_{k+l}] = [\underline{a}_0, ..., a_k] * [\underline{b}_0, ..., b_l]\,. \tag{2.38}$$

- The product of two decimal numbers $a_k a_{k-1} ... \underline{a}_0$ and $b_l a_{l-1} ... \underline{b}_0$ is the number $c_l c_{l-1} ... \underline{c}_0$ following from (2.38), except for the carrying. For example, we have $11 \times\ 11 = 121$ and $[1\ \underline{1}] * [1\ \underline{1}] = [1\ 2\ \underline{1}]$. This is the reason why, under certain conditions, the deconvolution of discrete signals can use division of numbers. For instance, we have $1331/11 = 121$ since $[1\ \underline{1}]^{-1} * [1\ 3\ 3\ \underline{1}] = [1\ 2\ \underline{1}]$. Since convolution also allows for negative numbers, this relation to multiplication is very practical; we easily can prove $[1\ -\underline{1}] * [-\underline{1}\ 1] = [1\ -\underline{2}\ 1]$.

2.4 Linear Systems Theory of Discrete Signals

Linear systems theory collects all properties, relations and operations for linear shift invariant signals. Convolution, as the basic operator on signals, is extended by a spectral decomposition, what is called the Fourier transformation, which allows to more deeply analyse the effect of linear filters but also the numerically efficient realization of convolution. We discuss the main results for discrete finite one-dimensional signals and then extend the results for the other types of signals and to two and more dimensions.

2.4.1 Convolution, Correlation, and Circulant Matrices

We start with the discrete convolution of finite vectors of length n. We assume the signal is periodic with period n, hence the indices are taken modulo n. Following (2.10), the indices run from $0, ..., n-1$, which simplifies expressions[7]. Convolution can be represented as a matrix-vector product by uniquely mapping each vector to a circulant matrix. This allows us to exploit results from linear algebra, especially the properties of eigenvalues and eigenvectors of a matrix.

2.4.1.1 Convolution and Correlation

We first want to represent convolution as a matrix-vector product.

The convolution of two n-vectors is defined as

$$g(i) = w(i) * f(i) = \sum_{k=0}^{n-1} w(i-k) f(i) \tag{2.39}$$

With the vectors

$$\boldsymbol{g} = [g_i] = \begin{bmatrix} g_0 \\ g_1 \\ \dots \\ g_{n-2} \\ g_{n-1} \end{bmatrix}, \qquad \boldsymbol{w} = [w_i] = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_{n-2} \\ w_{n-1} \end{bmatrix} \tag{2.40}$$

and the circulant matrix

$$\mathsf{Z}_f = \mathsf{Z}(\boldsymbol{f}) = \begin{bmatrix} f_0 & f_{n-1} & f_{n-2} & \dots & f_1 \\ f_1 & f_0 & f_{n-1} & \dots & f_2 \\ f_2 & f_1 & f_0 & \dots & f_3 \\ \dots & \dots & \dots & \dots & \dots \\ f_{n-1} & f_{n-2} & f_{n-3} & \dots & f_0 \end{bmatrix} \tag{2.41}$$

of the vector $\boldsymbol{f} = [f_i]$ we can write (2.39) as

$$\boldsymbol{g} = \boldsymbol{w} * \boldsymbol{f} = \mathsf{Z}_w \, \boldsymbol{f} = \mathsf{Z}_f \, \boldsymbol{w} \, . \tag{2.42}$$

Similarly, the correlation of two n-vectors, which is defined as

$$g(i) = w(i) \circledast f(i) = w(-i) * f(i) = \sum_{k=1}^{n-1} w(k) f(i+k) = \sum_{k=1}^{n-1} w(-k) f(i-k) \tag{2.43}$$

again, assuming all indices to be cyclical modulo n, also can be written with vectors and circulant matrices as

$$\boldsymbol{w} \circledast \boldsymbol{f} = \mathsf{Z}_w^{\mathsf{T}} \boldsymbol{f} \, , \tag{2.44}$$

since inversion of the indices leads to a transposed circulant matrix

$$\mathsf{Z}([w(-i)]) = \mathsf{Z}^{\mathsf{T}}([w(i)]) \, . \tag{2.45}$$

2.4.1.2 Correlation Function of Stochastic Signals

We often have to handle signals $\boldsymbol{f} = (f_i)$ which are assumed to be samples of a random process $\underline{\boldsymbol{f}} = (\underline{f}_i)$, where the random variables $\underline{f}_i$, which need not be mutually independent. The sequence (f_i) also is called a *stochastic process*. As an example, we may interpret the grey values $\boldsymbol{f}$ of an image line as a sample of a stochastical signal; then we can characterize the complete vector by its joint high-dimensional probability density function $p_f(\boldsymbol{f})$. If

stochastic process

[7] It is the reason why in the language C, developed by researchers working in signal theory, indices of vectors and matrices start with 0, not as in MATLAB, where the the first element in a vector has index 1

the statistical properties of $\underline{\boldsymbol{f}}$ are shift invariant, then – as we will see – their covariance matrix also is shift invariant. Its density may be represented by the mean $\mathbb{E}(\underline{f}_i) = \mu_f$ and the covariance $\mathbb{C}\text{ov}(\underline{f}_i, \underline{f}_j)$ of two values, which only depend on the difference $j - i$ of the two indices. If only these two moments of the distribution are given, following the principle of maximum entropy, see Sect. 1.6, p. 16, the most likely distribution of the signal is a Gaussian. For a detailed discussion of stochastic processes see (Papoulis, 1991), a short introduction is given in Förstner and Wrobel (2016, Sect. 2.8).

In the following we discuss how to arrive at estimated covariance matrices for signals with shift-invariant stochastical properties. These may be used to characterize stochastical signals. We will exploit this type of characterization when analysing image processing operations, especially noise suppression, and when describing the properties of textured areas in digital images.

Let two stochastical signals $\underline{\boldsymbol{f}}$ and $\underline{\boldsymbol{g}}$ be given. We assume, their mean is

$$\mathbb{E}(\underline{\boldsymbol{f}}) = \mathbb{E}(\underline{\boldsymbol{g}}) = \mathbf{0}\,. \tag{2.46}$$

We now determine the empirical covariance matrix $\mathbb{C}\text{ov}(\underline{\boldsymbol{f}}, \underline{\boldsymbol{g}})$. We obtain the (estimated) covariance.

$$\underset{\text{(independent on } i)}{\mathbb{C}\text{ov}(\underline{f}(i), \underline{g}(i))} = \frac{1}{n}\sum_{j=0}^{n-1} f(i+j)g(i+j) \overset{i=0}{=} \frac{1}{n}\sum_{j=0}^{n-1} f(j)g(j)\,, \tag{2.47}$$

taking all arguments modulo n. Observe, since we assumed the stochastical properties of the signal are shift invariant, the covariance of two values of $\underline{\boldsymbol{f}}$ and $\underline{\boldsymbol{g}}$ referring to the same index i are identical. In the area of signal processing this value often is called the *correlation*, also in case the mean values are not 0, hence, the correlation statistically is the second non-central moment.

The (estimated) covariance $c(i)$ after shifting $\boldsymbol{g}$ by what is called the *lag* τ then is *lag* τ

$$\begin{aligned}
\underset{\text{(independent on } i)}{c_{fg}(\tau) := \mathbb{C}\text{ov}(\underline{f}(i), \underline{g}(i+\tau))} &= \frac{1}{n}\sum_{j=0}^{n-1} f(i+j)\, g(i+j+\tau) && (2.48)\\
&\overset{i=0}{=} \frac{1}{n}\sum_{j=0}^{n-1} f(j)\, g(j+\tau) && (2.49)\\
&\overset{(2.43)}{=} \frac{1}{n} f(\tau) \circledast g(\tau)\,. && (2.50)
\end{aligned}$$

The function $c_{fg}(\tau)$ is the *empirical cross-covariance function*, or short, *cross-covariance function* of the signals $\underline{f}(i)$ and $\underline{g}(i)$. It only depends on the shift τ between the two signals. Again, in the area of signal processing, this function $c_{fg}(\tau)$ usually is called *correlation function* and named R_{fg}.

If both signals are periodic, then we obtain the circulant (and symmetric) covariance matrix, using the abbreviation $c = c_{fg}$ for brevity,

$$\begin{aligned}
\mathbb{C}\text{ov}(\underline{\boldsymbol{f}}, \underline{\boldsymbol{g}}) = \Sigma_{fg} = Z_c &= \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ c_2 & c_1 & c_0 & \dots & c_3 \\ \dots & \dots & \dots & \dots & \dots \\ c_{n-1} & c_{n-2} & c_{n-3} & \dots & c_0 \end{bmatrix} \\
&= \frac{1}{n}\begin{bmatrix} f_0 & f_1 & f_2 & \dots & f_{n-1} \\ f_{n-1} & f_0 & f_1 & \dots & f_{n-2} \\ f_{n-2} & f_{n-1} & f_0 & \dots & f_{n-3} \\ \dots & \dots & \dots & \dots & \dots \\ f_1 & f_2 & f_3 & \dots & f_0 \end{bmatrix} \begin{bmatrix} g_0 & g_{n-1} & g_{n-2} & \dots & g_1 \\ g_1 & g_0 & g_{n-1} & \dots & g_2 \\ g_2 & g_1 & g_0 & \dots & g_3 \\ \dots & \dots & \dots & \dots & \dots \\ g_{n-1} & g_{n-2} & g_{n-3} & \dots & g_0 \end{bmatrix} \\
&= \frac{1}{n} Z_f^T Z_g\,, && (2.51)
\end{aligned}$$

again, with cyclical indices. Observe, since we assumed the mean of both signals to be known to be 0, the denominator is n.

The autocorrelation[8] function (ACF) of a cyclical signal f hence is

$$c_{ff}(\tau) = \underset{\text{(independent on } i)}{\mathbb{C}\text{ov}}(\underline{f}(i), \underline{f}(i+\tau)) = \frac{1}{n}\sum_{t=0}^{n-1} f(t)f(t+\tau) = \frac{1}{n} f(\tau) \circledast f(\tau) \qquad \text{or} \qquad \boldsymbol{c}_{ff} = \frac{1}{n} Z_f^{\mathsf{T}} \boldsymbol{f} \tag{2.52}$$

Interpreted statistically, this is the autocovariance function of $\underline{f}(i)$ if the mean value is 0. This at the same time is the vector containing the variance (as first element) and the covariances to lag τ, i.e., the autocovariance function. It is also periodic, but in addition, it is also *symmetric*, i.e., we have (for real-valued signals)

$$c_{ff}(\tau) = c_{ff}(-\tau) \tag{2.53}$$

This easily can be proven using (2.51), via the general rule

$$\boldsymbol{g} = Z_g \boldsymbol{\delta} \quad \text{with} \quad \boldsymbol{\delta}^{\mathsf{T}} = [1,\ 0,\ 0, \ldots,\ 0]\,, \tag{2.54}$$

extracting the vector of the circulant matrix, which is the first column with index 0.

The results found so far can be summarized as following:

1. The convolution of two signals can be represented as

$$a(i) = b(i) * c(i) \quad \Leftrightarrow \quad \boldsymbol{a} = Z_b \boldsymbol{c} = Z_c \boldsymbol{b} \quad \Leftrightarrow \quad Z_a = Z_b Z_c = Z_c Z_b \tag{2.55}$$

   with the circulant matrix Z_b of the vector $\boldsymbol{b}$. The convolution is *commutative*.

2. The correlation of two signals can be represented as

$$a(i) = b(i) \circledast c(i) = b(-i) * c(i) \quad \Leftrightarrow \quad \boldsymbol{a} = Z_b^{\mathsf{T}} \boldsymbol{c} \quad \Leftrightarrow \quad Z_a = Z_b^{\mathsf{T}} Z_c \tag{2.56}$$

   The correlation is *not commutative*.

3. The unit matrix I_n corresponds to the unit vector $\boldsymbol{\delta}$. Hence n-vectors $\boldsymbol{a}$ with the addition and the convolution operators form a commutative ring $(\boldsymbol{a}, +, *)$ with $\mathbf{0}$ and $\boldsymbol{\delta}$ as identity elements, respectively, thus addition and convolution of signals can be used in the same way as addition and multiplication for real numbers $x \in \mathbb{R}$, which also form a commutative ring $(x, +, \cdot)$ with 0 and 1 as identity elements.

2.4.2 Spectral Decomposition of Circulant Matrices

The convolution in the form of the matrix-vector multiplication $\boldsymbol{a} = Z_b \boldsymbol{c}$ requires n^2 scalar multiplications and is inefficient for large n, unless the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ are sparse.

The complexity of the convolution can be reduced to $O(n \log n)$ when using the eigenvalue decomposition of circulant matrices. We have the following theorem.

Theorem 2.4.7: ***Eigenvalue decomposition of circulant matrices.*** For a n-vector $\boldsymbol{x} = [x_0, x_1, ..., x_{n-1}]$ with its circulant matrix

$$Z_x = [(Z_x)_{ik}] = [x_{i-k}]$$

we have the eigen decomposition

$$\boxed{Z_x = F^{-1}\text{Diag}(\boldsymbol{\lambda})F \quad \text{with} \quad \boldsymbol{\lambda} = F\boldsymbol{x}\,.} \tag{2.57}$$

The columns $\boldsymbol{f}_i$ of F, the eigenvectors, are invariant w.r.t. the vector $\boldsymbol{x}$ specifying the cyclical matrix Z_x. Using the n-th root of unity

$$w = \sqrt[n]{1} = \mathrm{e}^{\mathrm{i}2\pi/n}\,. \tag{2.58}$$

[8] We use the term used in signal processing.

the matrix of eigenvectors is the symmetric Fourier matrix

$$F = [f_{ik}] = [\mathrm{e}^{-\mathrm{i}2\pi ik/n}] = [w^{-ik}], \qquad \mathrm{i} = \sqrt{-1} \quad \text{and} \quad i,k \in [0,1,...,n-1]\,,$$

containing the eigenvectors

$$\boldsymbol{f}_i = [1\,, w^{-i}\,, ..., w^{-(n-1)i}]^{\mathsf{T}}\,. \tag{2.59}$$

The vector of eigenvalues, simply given by $\boldsymbol{\lambda} = F\boldsymbol{x}$, explicitly reads as

$$\lambda_i = \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{x} = \sum_{k=0}^{n-1} \mathrm{e}^{-\mathrm{i}2\pi ik/n} x_k\,. \tag{2.60}$$

Observe, the indices of vectors and matrices start with 0.

Example 2.4.12: ***Eigen decomposition of a circulant matrix.*** Let the 4-vector $\boldsymbol{x}$ and the circulant matrix Z_x, together with the 4×4 Fourier matrix F be

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \qquad Z_x = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}, \qquad F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -\mathrm{i} & -1 & \mathrm{i} \\ 1 & -1 & 1 & -1 \\ 1 & \mathrm{i} & -1 & -\mathrm{i} \end{bmatrix}. \tag{2.61}$$

The the diagonalization of Z_x as in (2.64) yields

$$FZ_xF^{-1} = \frac{1}{4}\begin{bmatrix} 10 & 10 & 10 & 10 \\ -2+2\mathrm{i} & 2+2\mathrm{i} & 2-2\mathrm{i} & -2-2\mathrm{i} \\ -2 & 2 & -2 & 2 \\ -2-2\mathrm{i} & 2-2\mathrm{i} & 2+2\mathrm{i} & -2+2\mathrm{i} \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \mathrm{i} & -1 & -\mathrm{i} \\ 1 & -1 & 1 & -1 \\ 1 & -\mathrm{i} & -1 & \mathrm{i} \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & -2+2\mathrm{i} & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -2-2\mathrm{i} \end{bmatrix}$$

The vector of the eigenvalues is

$$\boldsymbol{\lambda} = F\boldsymbol{x} = \begin{bmatrix} 10 \\ -2+2\mathrm{i} \\ -2 \\ -2-2\mathrm{i} \end{bmatrix}. \tag{2.62}$$

◇

This decomposition has the following properties:

- The columns of F^{-1} are the eigenvectors of the eigenvalue decomposition. The eigenvectors are independent of the vector $\boldsymbol{x}$.
- The matrix F is called the *Fourier matrix*.
- The matrix
$$\overline{F} = \frac{1}{\sqrt{n}}F \tag{2.63}$$
is unitary, i.e., $\overline{F}\overline{F}^* = \overline{F}^*\overline{F} = I_n$.
- Therefore we have the orthogonality relation
$$F^{-1} = \frac{1}{n}F^* \tag{2.64}$$
since $FF^* = nI_n$
- The elements of the vector $\boldsymbol{\lambda} = F\boldsymbol{x}$ are the eigenvalues of Z_x. If Z_x is not symmetric, they are complex numbers. Since the matrix $\overline{F}$ is orthonormal, it can be interpreted as a rotation matrix.

Remark: There exist alternative definitions of the Fourier transformation which differ by a factor:

- Gonzales and Wintz (1977) define the Fourier transformation as
$$\boldsymbol{X} = \frac{1}{n}F\boldsymbol{x}\,, \qquad \boldsymbol{x} = F^*\boldsymbol{X} \tag{2.65}$$

- Castleman (1996) defines it symmetrical

$$\boldsymbol{X} = \overline{F}\boldsymbol{x}\,, \qquad \boldsymbol{x} = \overline{F}^*\boldsymbol{X}\,. \tag{2.66}$$

When generalizing the Fourier transformation to continuous nonperiodic signals we will adopt the symmetric definition. ◇

2.4.3 The Discrete Fourier Transformation

The transformation

$$\boldsymbol{X} = F\boldsymbol{x} \tag{2.67}$$

is called the *discrete Fourier transformation*. The DFT thus is the transformation of a vector into the eigenvalues of the corresponding circulant matrix. Since with (2.63) it can be written as

$$\boldsymbol{X} = \sqrt{n}\,\overline{F}\boldsymbol{x}\,. \tag{2.68}$$

Hence, the *Fourier transformation is a rotation* except for a constant factor $\sqrt{n}$. Observe, the original signal is written with small letters and will have indices i, its Fourier transform is written with capital letters and will have indices k.[9]

We explicitly have

$$X_k = \sum_{i=0}^{n-1} x_i\, \mathrm{e}^{-\mathrm{i}2\pi ik/n} = \sum_{i=0}^{n-1} x_i\, (\cos(2\pi ik/n) - \mathrm{i}\sin(2\pi ik/n)) \tag{2.69}$$

and also

$$\Re(X_k) = \sum_{i=0}^{n-1} x_i\, \cos(2\pi ik/n) \qquad \Im(X_k) = -\sum_{i=0}^{n-1} x_i\, \sin(2\pi ik/n)\,, \tag{2.70}$$

where the operators $\Re$ and $\Im$ extract the real and the imaginary part of a complex number, respectively.

The DFT is invertible. The inverse discrete Fourier transformation (IDFT) is given by

$$\boldsymbol{x} = F^{-1}\boldsymbol{X} = \frac{1}{n}F^*\boldsymbol{X} \tag{2.71}$$

This can be written explicitly in various forms. Using Euler's relation $\mathrm{e}^{\mathrm{i}\alpha} = \cos\alpha + \mathrm{i}\sin\alpha$ we first have

$$x_i = \frac{1}{n}\sum_{k=0}^{n-1} X_k\, \mathrm{e}^{\mathrm{i}2\pi ik/n} \tag{2.72}$$

$$= \frac{1}{n}\sum_{k=0}^{n-1} X_k(\cos(2\pi ik/n) + \mathrm{i}\sin(2\pi ik/n))\,. \tag{2.73}$$

Since the x_i are reals, sin-terms in (2.73) need to vanish. Hence, we have $X_k = X^*_{n-k}$, or $\Re(X_k) = \Re(X_{n-k})$ and $\Im(X_k) = -\Im(X_{n-k})$. Therefore, only n real values are necessary to represent the Fourier spectrum. Therefore, we obtain the following relation Exercise 2.4

$$x_i = \frac{2}{n}\sum_{k=0}^{n-1} \Re(X_k)\cos(2\pi ik/n) - \Im(X_k)\sin(2\pi ik/n) \tag{2.74}$$

Finally we represent X_k in polar coordinates

$$X_k = |X_k|\,\mathrm{e}^{\mathrm{i}\varphi_n} = |X_k|\,(\cos\varphi_n + \mathrm{i}\sin\varphi_n)\,. \tag{2.75}$$

[9] The Fourier transform of $\boldsymbol{x}$ often is written as $\hat{\boldsymbol{x}}$, see Jähne et al. (1999). We do not follow this convention, since we reserve the hat for estimates.

With the trigonometric addition theorem, we finally obtain

$$x_i \quad = \quad \frac{2}{n} \sum_{k=0}^{n-1} |X_k| \cos(2\pi ik/n + \varphi_n) \,. \tag{2.76}$$

an expression which makes the decomposition of the signal (x_i) in periodic summands with their amplitude $|X_k$ and their phase φ_k explicit.

We have the following notions

- The sequence (X_k) is the Fourier spectrum of the sequence (x_i) and
- the sequence $(|X_k|)$ is the amplitude spectrum of (x_i).

The DFT and its inverse, the IDFT

$$\boldsymbol{X} = \mathsf{F}\boldsymbol{x} \qquad \boldsymbol{x} = \mathsf{F}^{-1}\boldsymbol{X} = \frac{1}{n}\mathsf{F}^*\boldsymbol{X} \,, \tag{2.77}$$

often are denoted as

$$\boldsymbol{x} \circ\!\!-\!\!\bullet \, \boldsymbol{X} \,, \tag{2.78}$$

or more explicitly with the operator $\mathcal{F}$

$$\boldsymbol{X} = \mathcal{F}(\boldsymbol{x}) \quad \text{and} \quad \boldsymbol{x} = \mathcal{F}^{-1}(\boldsymbol{X}) \,, \tag{2.79}$$

acting on the sequence, the function or the vector, depending on the representation. The vectors $\boldsymbol{x}$ and $\boldsymbol{X}$ are called a Fourier pair.

The result of this section can be summarized as follows:

- The eigenvalue spectrum of a circulant matrix Z_x with vector $\boldsymbol{x} = [x_i]$ is identical to the Fourier-Spectrum $\boldsymbol{X} = [X_k]$ of the periodic sequence (x_i).
- The absolute values $|X_k|$ of the eigenvalues of Z_x are the amplitude spectrum of the periodic sequence (x_i).

Example 2.4.13: ***Amplitude Spectrum of discrete periodic signal.*** Figure 2.9 shows the signal $x(i)$ from Fig. 2.5 and approximations together with its amplitude spectra. Observe, the elements of the Fourier spectrum have indices $k \in [-8, +8]$ and the elements with high values of $|k|$ are set to zero in order to obtain the approximations. ◇

2.4.4 The Power spectrum

We can apply this type of spectral analysis to the autocorrelation function c_x. As shown in (2.52), the autocorrelation function of a signal x

$$c_x(i) = \frac{1}{n} \sum_{i'} x(i')x(i+i') \tag{2.80}$$

contains the empirical covariances of the signal, assuming it has zero mean. It is symmetric, see (2.53). The autocovariance function of a signal x sometimes is called R_x.

The corresponding definition of the autocovariance function using circulant matrices

$$\frac{1}{n}\mathsf{Z}_x^T\mathsf{Z}_x = \mathsf{Z}_c \,, \tag{2.81}$$

by multiplying from left and right with F and F^{-1}, and augmenting by $\mathsf{F}^{-1}\mathsf{F}$, can be rewritten as

$$\frac{1}{n}\mathsf{F}\mathsf{Z}_x^T\mathsf{F}^{-1}\mathsf{F}\mathsf{Z}_x\mathsf{F}^{-1} = \mathsf{F}\mathsf{Z}_c\mathsf{F}^{-1} \,. \tag{2.82}$$

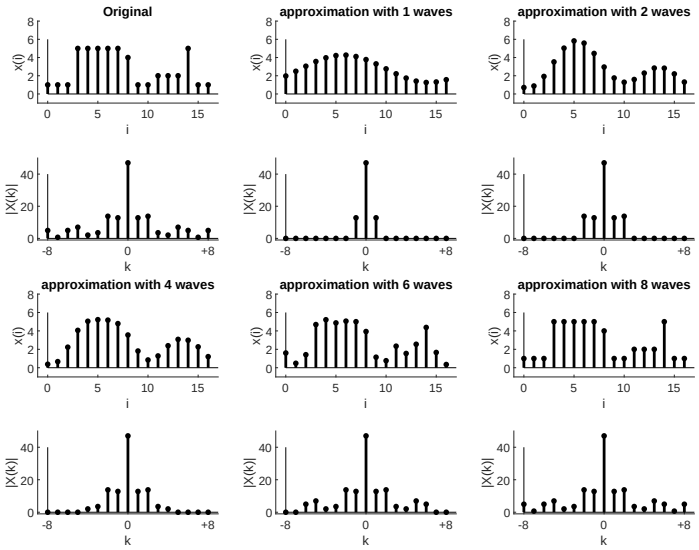


Figure 2.9: Signal $x(i)$ from Fig. 2.5 and approximations with amplitude spectra $|X(k)|$

This is a product of diagonal matrices having diagonal elements

$$\overline{\boldsymbol{X}} = \boldsymbol{F}\overline{\boldsymbol{x}}\,, \quad \boldsymbol{X} = \boldsymbol{F}\boldsymbol{x}\,, \quad \text{and} \quad \boldsymbol{C} = \boldsymbol{F}\boldsymbol{c} \tag{2.83}$$

where $\overline{\boldsymbol{x}} = [x_0, x_{n-1}, ..., x_1]$ is the reverse signal. Therefore we have the pointwise (Hadamard) product

$$\frac{1}{n}\overline{\boldsymbol{X}} \circ \boldsymbol{X} = \boldsymbol{C}\,, \tag{2.84}$$

where $\overline{\boldsymbol{X}}$ is the Fourier transform of $\overline{\boldsymbol{x}}$. Since $\overline{\boldsymbol{X}} = \boldsymbol{X}^*$, for a single element we have $\frac{1}{n}X_k^* X_k = C_k$ or Exercise 2.5

$$C_k = \frac{1}{n}|X_k|^2\,. \tag{2.85}$$

Hence, the Fourier spectrum $\boldsymbol{C}$ of the covariance function $\boldsymbol{c}$ is identical to the squared of the amplitude spectrum $|X_k|$ of the signal $\boldsymbol{x}$, except for a factor $1/n$. The function C often is called the power spectrum of the signal x. Hence, we have

$$c \circ\!\!-\!\!\bullet\, nC\,. \tag{2.86}$$

In signal processing the power spectrum often is called P.

The Fourier spectrum $\boldsymbol{C}$ of the covariance function c is real valued and ≥ 0. Therefore, we have, what is called the *Wiener-Kinchin relation*:

$$C_k = \frac{1}{n}\sum_k c_i \cos(2\pi ik/n) \tag{2.87}$$

$$c_i = \sum_k C_k \cos(2\pi ik/n)\,. \tag{2.88}$$

It holds for any signal which is symmetric, i.e., if $x(i) = x(-i)$. Then the (2.87) is also called the cosine-transformation, and (2.88) the inverse cosine-transformation.

The variance is

$$\sigma^2 = \frac{1}{n}\sum_i x_i^2 = \frac{1}{n}\boldsymbol{x}^T\boldsymbol{x} \tag{2.89}$$

and using the orthogonality relation (2.64), p. 34 directly leads to what is called the *Parseval identity*

$$\boldsymbol{X}^*\boldsymbol{X} = n\boldsymbol{x}^T\boldsymbol{x} \tag{2.90}$$

or

$$\sum_k |X_k|^2 = n\sum_i x_i^2\,. \tag{2.91}$$

The estimated variance of a sample signal $\boldsymbol{x}$, taken from $\underline{\boldsymbol{x}} \sim \mathcal{M}(\boldsymbol{0}, \mathbb{D}(\underline{\boldsymbol{x}}))$ with $\mathbb{D}(\underline{\boldsymbol{x}}) = \boldsymbol{Z}_c$ therefore is

$$\widehat{\sigma^2} = \frac{1}{n}\sum_i x_i^2 = \frac{1}{n^2}\sum_k |X_k|^2 = \frac{1}{n}\sum_k \widehat{C}_k\,. \tag{2.92}$$

If the signal $(\underline{x}_i)$ is stochastic, the Fourier spectrum $\underline{\boldsymbol{X}} = \boldsymbol{F}\underline{\boldsymbol{x}}$ is stochastic and contains uncorrelated values. Their covariance matrix [10] is Exercise 2.6

$$\mathbb{D}(\underline{\boldsymbol{X}}) = \boldsymbol{\Sigma}_{XX} = \boldsymbol{F}\mathbb{D}(\underline{\boldsymbol{x}})\boldsymbol{F}^{-1} = n\mathrm{Diag}(\boldsymbol{F}\boldsymbol{c}) = n\mathrm{Diag}(C_k)\,. \tag{2.93}$$

with the elements C_k of the power spectrum as variances. Therefore, due to (2.76) we can interpret the power spectrum: it indicates how the variance of the signal is distributed to the individual wavelengths k. Observe the different representations for the covariance

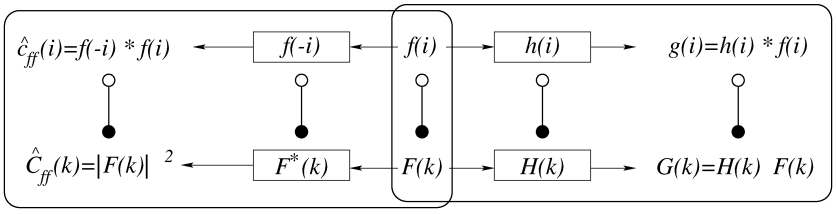


Figure 2.10: Relation between a signal $f(i)$, its estimated autocorrelation function $\widehat{c}_{ff}(i) = R_f(i)$, a filter $h(i)$ and the filtered signal $g(i)$ with their Fourier transforms $F(k)$, $\widehat{C}_{ff}(k) = \widehat{P}_f(k)$, $H(k)$ and $G(k)$ The **upper row** describes the convolutions of $f(i)$ with some filter $h(i)$ (**right**) to obtain a filtered signal $g(i)$ and with $f(-i)$ (**left**) to obtain the estimated autocorrelation function $\widehat{c}_{ff}(i) = f(-i) * f(i) = f(i) \circledast f(i)$. The **second row** describes the corresponding Fourier transformations. They are one-to-one mappings of the functions used in the first row. The Fourier transformation of the mirrored signal $f(-i)$ is the conjugate complex of $F(k)$. Convolution in the Fourier domain is realized by pointwise multiplication. The pointwise multiplication of the Fourier transformation $F(k)$ with its conjugate complex $F^*(k)$ yields the magnitude square $|F(k)|^2$ of the elements, which is the estimated power spectrum $\widehat{C}_{ff}$ of f (**left**)

matrix of the (estimated) power spectrum in (2.93). If the signal is deterministic, so also its Fourier transform.

The relations derived so far, can be collected in the left box of the diagram in Fig. 2.10. They are valid for all generalizations of the Fourier transformation to continuous and to non-periodic signals.

2.4.5 The Fast Fourier Transformation

The Fourier transform $\boldsymbol{X} = \mathsf{F}\boldsymbol{x}$ of an n-vector $\boldsymbol{x}$ requires n^2 multiplications. The structure of the Fourier matrix can be exploited to speed up the computation. The computational complexity of the Fourier transformation then is $O(n \log n)$. For n being a power of 2, the idea, is to split the signal into the values with odd and even index, apply the Fourier transformation to the halved sequence, and merge the result following Cooley and Tukey (1965). For other non-primes such a splitting is also possible, see Winograd (1976). Rader (1968) proposed a $O(n \log n)$-algorithm for n being a prime. The corresponding algorithms are called fast Fourier transformations (FFT), their inverses are denoted with IFFT.

Fast Fourier transformation plays a central role, since, as we saw, the convolution $\boldsymbol{c}$ of two signals $\boldsymbol{a}$ and $\boldsymbol{b}$ can be realized as the point-wise multiplication of the corresponding Fourier transforms. Since the Fourier transformation for large n is fast it may be useful to proceed as follows:

$$\boldsymbol{A} = \mathrm{FFT}(\boldsymbol{a})\,, \quad \boldsymbol{B} = \mathrm{FFT}(\boldsymbol{b})\,, \quad \boldsymbol{C} = \boldsymbol{A} \circ \boldsymbol{B}\,, \quad \boldsymbol{c} = \mathrm{IFFT}(\boldsymbol{C}) \tag{2.94}$$

. The complexity of this realization of the convolution is $O(n \log n)$.

Example 2.4.14: ***Fourier transformation of a 2- and a 4-vector.*** The Fourier transform of a 2-vector is

$$X_0 = x_0 + x_1 \tag{2.95}$$
$$X_1 = x_0 - x_1\,. \tag{2.96}$$

The fast Fourier transformation recursively uses the following procedure: Assuming n is even, the Fourier transform $\boldsymbol{X}$ of an n-vector can be derived from the Fourier transforms $\boldsymbol{E}$ and $\boldsymbol{O}$ of the vectors $[x_0, x_2, ...]^\mathsf{T}$ and $[x_1, x_3, ...]$, containing the values with even and odd indices, by

$$X_k = E_k + O_k \mathrm{e}^{-2\pi \mathrm{i} k/n}\,, \quad k = 0, ..., n/2 - 1\,, \tag{2.97}$$
$$X_{k+n/2} = E_k - O_k \mathrm{e}^{-2\pi \mathrm{i} k/n}\,, \quad k = 0, ..., n/2 - 1\,. \tag{2.98}$$

[10] The variance of a complex random variable $\underline{z} = \underline{x} + \mathrm{i}\underline{y}$ is defined as $\mathbb{D}(\underline{z}) = \mathbb{E}(|\underline{z} - \mathbb{E}(\underline{z})|^2) = \mathbb{D}(\underline{x}) + \mathbb{D}(\underline{y})$. The covariance of two complex variables $\underline{z}$ and $\underline{w}$ is defined as $\mathbb{C}\mathrm{ov}(\underline{z}, \underline{w}) = \mathbb{E}\left((\underline{z} - \mathbb{E}(\underline{z}))(\underline{w} - \mathbb{E}(\underline{w}))^*\right)$.

First we have (now for $n = 4$)

$$\boldsymbol{E} = \begin{bmatrix} x_0 + x_2 \\ x_0 - x_2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{O} = \begin{bmatrix} x_1 + x_3 \\ x_1 - x_3 \end{bmatrix} \tag{2.99}$$

With $\mathrm{e}^{-2\pi \mathrm{i} k/4} = -\mathrm{i}^k$ we obtain

$$\boldsymbol{X} = \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} (x_0 + x_2) + (x_1 + x_3) \\ (x_0 + x_2) - \mathrm{i}(x_1 + x_3) \\ (x_0 - x_2) + (x_1 - x_3) \\ (x_0 - x_2) - \mathrm{i}(x_1 - x_3) \end{bmatrix}, \tag{2.100}$$

see (2.61). ◇

2.4.6 The Two-dimensional Discrete Fourier Transform

The Fourier transform of a discrete two-dimensional function $g(i, j)$ results from

$$\begin{aligned} G(k,l) &= \sum_i \sum_j g(i,j) \mathrm{e}^{-\mathrm{i}2\pi(ik+jl)} \\ &= \sum_i \left(\sum_j g(i,j) \mathrm{e}^{-\mathrm{i}2\pi jl/n} \right) \mathrm{e}^{\mathrm{i}2\pi ik/n} \qquad (2.101) \\ &= \mathcal{F}_1(\mathcal{F}_2(g(i,j))), \qquad (2.102) \end{aligned}$$

the index of $\mathcal{F}$ indicating the number of the argument of the function g. It can be realised in two steps

1. Fourier transformation of all columns of $\gamma(i, l) := \mathcal{F}_2(g(i, j))$ which leads to a two-dimensional intermediate matrix $\boldsymbol{\gamma} = [\gamma(i, l)]$;
2. Fourier transformation of all rows of $G(k, l) = \mathcal{F}_1(\boldsymbol{\gamma}(i, l))$.

If we use the Fourier matrix F for realizing the 1D Fourier transformation, we thus obtain the 2D Fourier transformation of the matrix g

$$\mathsf{G} = \mathsf{F}\mathsf{g}\mathsf{F}^{\mathsf{T}}. \tag{2.103}$$

Hence, the two-dimensional Fourier transformation has the same complexity as the one-dimensional Fourier transformation. The same holds for the inverse Fourier transformation.

Example 2.4.15: ***JPEG Image Compression.***

The transformations (2.87) and (2.88) represent the discrete cos-Transformation (DCT) of the signal $c(i)$ and its inverse(IDCT). It is used within JPEG compression.

Here, image blocks of size 8×8 are mirrored in x- and y-direction, and assumed that they are continued periodically. This leads to the 16×16 block

$$b_c(i', j') = \begin{bmatrix} b(i,j) & b(-i,j) \\ b(i,-j) & b(-i,-j) \end{bmatrix} \qquad i, j \in [0, ..., 7],\ i', j' \in [0, ..., 15] \tag{2.104}$$

Then the two-dimensional DCT is applied to $b_c(i', j')$. From the real valued coefficients only those are stored which are larger that a threshold. Hence, all those frequencies are suppressed, whose amplitude are below this threshold. It is presumed, that these do are essential for visual perception. Decoding is realized by applying an inverse discrete cosine-transformation to the compressed signal, yielding smoothed versions of the original image blocks. Though the intensities within a block are smoothed, the coding is done for each block independently: this leaves discontinuities at the borders of the blocks, which can be perceived as the typical JPEG-compression artifacts.

◇

2.5 Generalizations

The spectral analysis using the DFT referred to discrete periodic signals. It can be generalized to non-periodic and continuous signals. We have the following transformations, referring to discrete and continuous function $f(i)$ and $f(x)$:

1. The "discrete Fourier transformation", discussed so far, is the Fourier transform of periodic discrete signals. Together with its inverse it is given by:

$$X(k) = \sum_{i=0}^{n-1} f(i)\mathrm{e}^{-\mathrm{i}2\pi ik/n} \quad \text{and} \quad f(i) = \frac{1}{n}\sum_{k=0}^{n-1} F(k)\mathrm{e}^{+\mathrm{i}2\pi ik/n}\,, \tag{2.105}$$

   where all indices are to be taken modulo n. Here the Fourier transform is also discrete and periodic.

2. "Fourier series" are the Fourier transform of periodic continuous signals. For signals with period L we have the Fourier series and its inverse

$$F(k) = \frac{1}{L}\int_{x=0}^{L} f(x)\mathrm{e}^{-\mathrm{i}2\pi kx/L}\mathrm{d}t \quad \text{and} \quad f(x) = \sum_{k=-\infty}^{\infty} F(k)\mathrm{e}^{+\mathrm{i}2\pi kx/L} \tag{2.106}$$

   Here the Fourier transform is discrete and non-periodic.

3. The "discrete time Fourier transform" is the Fourier transformation of non-periodic discrete signals $f(i\Delta x)$ and obtained from the Fourier transformation of continuous periodic signals by exchanging the role of signal and transform. Together with its inverse it is

$$F(u) = \sum_{n=-\infty}^{\infty} f(n\Delta x)\mathrm{e}^{-\mathrm{i}2\pi un\Delta x} \quad \text{and} \quad f(i) = \Delta x\int_{u=0}^{1/\Delta x} F(u)\mathrm{e}^{+\mathrm{i}2\pi ui\Delta x}\mathrm{d}u \tag{2.107}$$

   The Fourier transform is continuous and periodic, with period $1/\Delta x$. We will use it for analysing the properties of discrete filters.

4. The "Fourier transform" of a nonperiodic continuous signals together with its inverse is given by

$$F(u) = \int_{x=-\infty}^{\infty} f(x)\mathrm{e}^{-\mathrm{i}2\pi ux}\mathrm{d}x \quad \text{and} \quad f(x) = \int_{u=-\infty}^{\infty} F(u)\mathrm{e}^{+\mathrm{i}2\pi ux}\mathrm{d}u \tag{2.108}$$

   Observe, we used the symmetric definition, as in (Castleman, 1996). Since the Fourier transformation and its inverse just differ by the sign in the exponent, we have the following duality relation:

$$f(x) \circ\!\!-\!\!\bullet\, F(u) \quad \leftrightarrow \quad F(x) \circ\!\!-\!\!\bullet\, f(-u)\,. \tag{2.109}$$

   This relation is of advantage to derive the Fourier transform of a function which has the form of $F(u)$, if $f(x) \circ\!\!-\!\!\bullet\, F(u)$ is known.

We will discuss the continuous Fourier transformation of special 1D and 2D functions.

2.5.1 The 1D Fourier Transformation

The Fourier transformation of non-periodic continuous functions contains the other cases (discrete, or periodic functions) as special cases. The Fourier transformation $F(u)$, as defined in (2.108), p. 41, exists if (1) the function $f(x)$ is an quadratically integrable function, i.e., the integral $\int_{x=-\infty}^{\infty} f^2(x)\mathrm{d}x < \infty$. The definition of a Fourier transformation can be extended to functions, which are not quadratically integrable. All mentioned transformations can be interpreted as eigenvalue decompositions. [11]

[11] This is because the transformations can be embedded in the theory of Hilbert spaces.

We need Fourier transforms of a number of basic functions. The Fourier transform of a Gaussian function is a Gaussian function, see Fig. 2.11

$$\frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \circ\!\!-\!\!\bullet\, \mathrm{e}^{-2\pi^2\sigma^2u^2} \,. \tag{2.110}$$

Observe, a steeper Gaussian in the spatial domain leads to a broader Gaussian in the spectral domain.

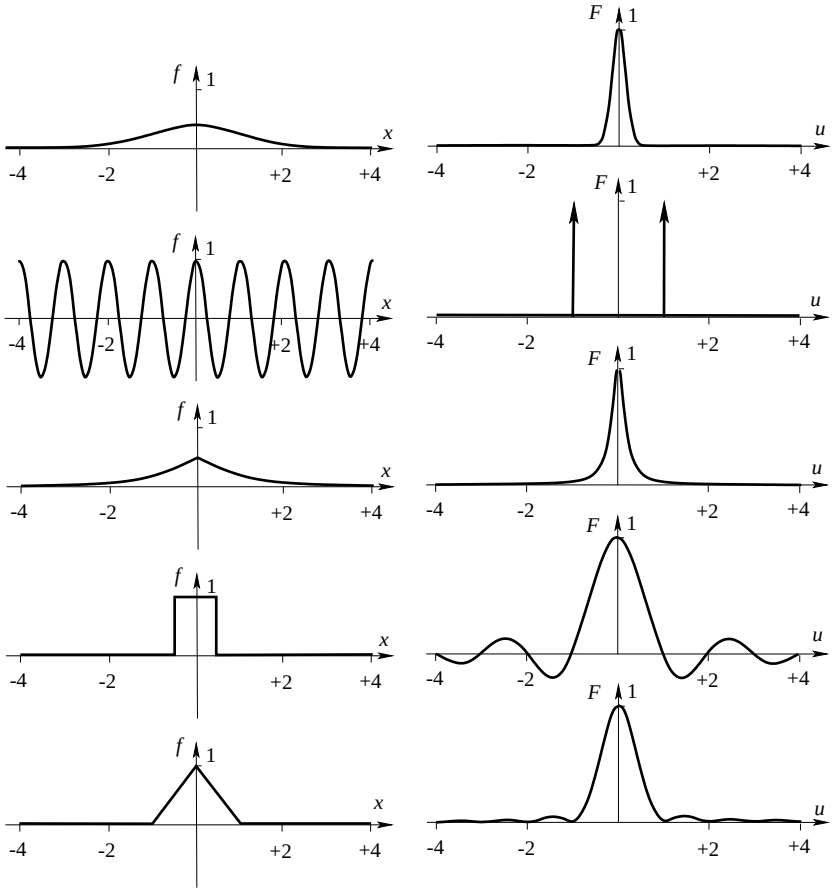


Figure 2.11: Examples for Fourier pairs. From top to bottom: Gaussian/Gaussian, cosine/double delta, doubly exponential/Cauchy, rectangle/sinc, hat/sinc^2. The delta function $\delta(x - x_0)$ is shown as peak at x_0, see the definition in Sect. 2.6.1.1, p. 45

The delta function is defined as the limit of the Gaussian with $\sigma \to \infty$, see 2.6, p. 44. Therefore, we have

$$\delta(x) \circ\!\!-\!\!\bullet\, 1 \quad \text{and} \quad 1 \circ\!\!-\!\!\bullet\, \delta(u) \,. \tag{2.111}$$

If we shift the delta function by x_0 we obtain

$$\delta(x - x_0) \circ\!\!-\!\!\bullet\, \mathrm{e}^{-\mathrm{i}2\pi x_0 u} \quad \text{and} \quad \mathrm{e}^{\mathrm{i}2\pi u_0 x} \circ\!\!-\!\!\bullet\, \delta(u - u_0) \,. \tag{2.112}$$

This relation can be used to derive the Fourier transform of a shifted function, since $f(x - x_0) = f(x) * \delta(x - x_0)$; hence

$$f(x - x_0) \circ\!\!-\!\!\bullet\, \mathrm{e}^{-\mathrm{i}2\pi x_0 u} F(u) \,. \tag{2.113}$$

Similarly, we can derive the Fourier transform of a scaled function

$$f(ax) \circ\!\!-\!\!\bullet\, \frac{1}{|a|} F\left(\frac{u}{a}\right) \,. \tag{2.114}$$

Thus, if $|a| < 1$ the function is wider, becomes flatter and smoother , and its Fourier transform is more concentrated around 0. Using the right relation of (2.112) and $\cos x = \frac{1}{2}(\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x})$, we easily can derive

$$\cos(2\pi u_0 x) \quad \circ\!\!-\!\!\bullet \quad \frac{1}{2}[\delta(u+u_0) + \delta(u-u_0)]\,, \tag{2.115}$$

$$\sin(2\pi u_0 x) \quad \circ\!\!-\!\!\bullet \quad \frac{\mathrm{i}}{2}[\delta(u+u_0) - \delta(u-u_0)]\,. \tag{2.116}$$

The Fourier transform of the n-th derivative $f^{(n)}(x)$ of a function $f(x)$ yields

$$f^{(n)}(x) \circ\!\!-\!\!\bullet (\mathrm{i}2\pi u)^n F(u)\,. \tag{2.117}$$

The Fourier transform of the doubly exponential function $\mathrm{e}^{-a|x|}$ is proportional to the density of the central Cauchy-distribution $p(x;a) = a/(a^2+x^2)/\pi$. We have the Fourier transform together with its dual

$$\mathrm{e}^{-a|x|} \circ\!\!-\!\!\bullet \frac{2a}{a^2 + 4\pi^2 u^2} \quad \text{and} \quad \frac{1}{\pi}\frac{a}{a^2+x^2} \circ\!\!-\!\!\bullet \mathrm{e}^{-2\pi a|u|}\,. \tag{2.118}$$

The unit-rectangle function has the sinc-function as its Fourier transformation

$$r\left(x; -\frac{1}{2}, \frac{1}{2}\right) \circ\!\!-\!\!\bullet \operatorname{sinc} u := \frac{\sin\,\pi u}{\pi u}\,. \tag{2.119}$$

The hat-function $\Lambda(x;-1,1) = r\left(x;-\frac{1}{2},\frac{1}{2}\right) * r\left(x;-\frac{1}{2},\frac{1}{2}\right)$ is the convolution of the box-function with itself. Hence its Fourier transform is the square of the sinc-function:

$$\Lambda\left(x; -\frac{1}{2}, \frac{1}{2}\right) \circ\!\!-\!\!\bullet \operatorname{sinc}^2 u = \frac{\sin^2\,\pi u}{(\pi u)^2}\,. \tag{2.120}$$

Generally, we can describe the statistical properties of signals using their autocorrelation function $R_f(\tau)$. This is relevant for characterizing the texture in an image or when analysing the accuracy of image matching procedures. A prominent feature is the curvature of the autocorrelation function $R_f''(\tau)$ for lag $\tau = 0$, see p. 32. We have the following result: The curvature of the autocorrelation function at lag $\tau = 0$ is the negative variance of the first derivative of the signal:

curvature of autocorrelation is negative variance of gradient

$$R_f''(0) = -\sigma_{f_x}^2\,. \tag{2.121}$$

This can be used to characterize the autocorrelation function with one parameter. High curvature corresponds to a rough signal, low curvature to a smooth signal. A generalization of the variance $\sigma_{f_x}^2$ of the first derivative to two dimensions results in what is called the structure tensor, which plays a significant role in image matching and keypoint detection.

Proof: We need the autocorrelation function $R_{f_x}(\tau)$ of the first derivative $f_x(x) = \mathrm{d}f(x)/\mathrm{d}x$ of a function $f(x)$, see (Papoulis, 1965, p. 316 ff.). Provided the process is smooth enough, such that the derivatives exist and their mean is zero, we have due to the linearity of differentiation

$$\mathbb{E}\left(\frac{\mathrm{d}f(x)}{\mathrm{d}x}\frac{\mathrm{d}f(y)}{\mathrm{d}y}\right) = \frac{\mathrm{d}^2\mathbb{E}(f(x)f(y))}{\mathrm{d}x\mathrm{d}y}\,. \tag{2.122}$$

If the lag $\tau = y - x$ is constant, hence $y = \tau + x$ and $x = y - \tau$ we therefore obtain

$$R_{f_x}(\Delta x) := \mathbb{E}\left(\frac{\mathrm{d}f(y-\tau)}{\mathrm{d}\tau}\frac{\mathrm{d}f(x+\tau)}{\mathrm{d}\tau}\right) = -\frac{\mathrm{d}^2\mathbb{E}(f(x)f(y))}{\mathrm{d}\tau^2} = -\frac{\partial^2 R_f(\tau)}{\mathrm{d}\tau^2}\,. \tag{2.123}$$

Hence for lag $\tau = 0$ we get (2.121). ◇

Visualizing the 1D Fourier transform, which in general is complex valued $F(u) = |F(u)|\mathrm{e}^{\mathrm{i}\phi(u)}$, is done in several ways:

- The separate Visualization of amplitude $|F(u)|$and phase $\phi(u)$, see Fig. 2.12, left;
- The visualization as a complex function $[\Re(F(u)), \Im(F(u))]$ in one variable u in a 3D coordinate system, see Fig. 2.12, middle.
- The projection of this function into the complex plane, indicating points with distinct u, see Fig. 2.12, right.

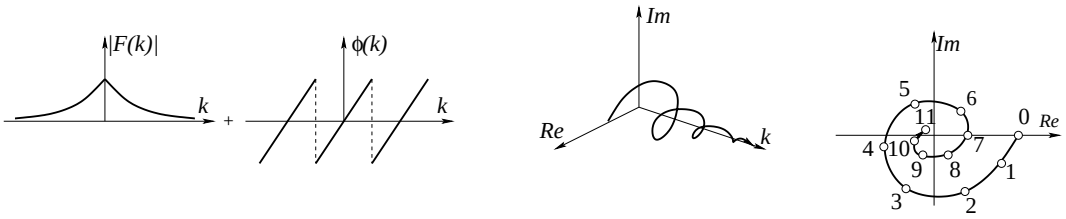


Figure 2.12: Visualization of the Fourier transform

2.5.2 The 2D Fourier Transformation

The two-dimensional Fourier transformation is defined analogously, see (Castleman, 1996). For simplifying notation, we use vector valued arguments, which partially allows us to generalize to more than two dimensions. We have

$$F(\boldsymbol{u}) = \mathcal{F}(f(\boldsymbol{x})): \qquad F(\boldsymbol{u}) = \int_{\boldsymbol{x}} f(\boldsymbol{x}) \mathrm{e}^{-\mathrm{i}2\pi \boldsymbol{u}^T \boldsymbol{x}} \mathrm{d}\boldsymbol{x}\,. \tag{2.124}$$

The inverse Fourier transformation is

$$f(\boldsymbol{x}) = \mathcal{F}^{-1}(F(\boldsymbol{u})): \qquad f(\boldsymbol{x}) = \int_{\boldsymbol{u}} F(\boldsymbol{u}) \mathrm{e}^{\mathrm{i}2\pi \boldsymbol{x}^T \boldsymbol{u}} \mathrm{d}\boldsymbol{u}\,, \tag{2.125}$$

therefore

$$f(\boldsymbol{x}) \circ\!\!-\!\!\bullet F(\boldsymbol{u}) \tag{2.126}$$

If we linearly transform $\boldsymbol{x}$ by A we obtain for the Fourier transform of $f(A\boldsymbol{x})$

$$f(A\boldsymbol{x}) \circ\!\!-\!\!\bullet \frac{1}{\det A} F((A^T)^{-1}\boldsymbol{u}) \tag{2.127}$$

which specializes for diagonal $D = \mathrm{Diag}([d_1, d_2])$

$$f(D\boldsymbol{x}) \circ\!\!-\!\!\bullet \frac{1}{d_1 d_2} F(D^{-1}\boldsymbol{u}) \tag{2.128}$$

and for a rotation matrix R

$$f(R\boldsymbol{x}) \circ\!\!-\!\!\bullet F(R\boldsymbol{u})\,. \tag{2.129}$$

Hence a widening of $f(\boldsymbol{x})$ ($|D| < 1$) leads to a narrowing of the Fourier transform together with an enlargement of its values. Hence, the rotation of an image just rotates its Fourier transform. A shift of $f(\boldsymbol{x})$ by $\boldsymbol{a}$ leads to

$$f(\boldsymbol{x} + \boldsymbol{a}) \circ\!\!-\!\!\bullet \mathrm{e}^{-\mathrm{i}2\pi \boldsymbol{u}^T \boldsymbol{a}} F(\boldsymbol{u}) \tag{2.130}$$

hence only changes the phase.

The visualization of the 2D Fourier transform usually is restricted to the absolute value, i.e., the amplitude spectrum $|F(\boldsymbol{u})|$. Due to the quick drop-off of $|F(\boldsymbol{u})|$ often $\log(1 + |F(\boldsymbol{u})|)$ is shown.

2.6 Sampling and Interpolation

Usually we work with sampled functions. Thus, we take the image plane as a continuum where, via sensor elements, the continuous image function is sampled, usually in a regular grid. In this section we want to discuss the conditions under which the sample of a continuous function allows to reconstruct the function from the sample. This leads to the sampling theorem. We obviously need to formally describe the two subsequent processes, the sampling and the interpolation. Both can be adequately described by convolutions, thus exploit the material discussed so far.

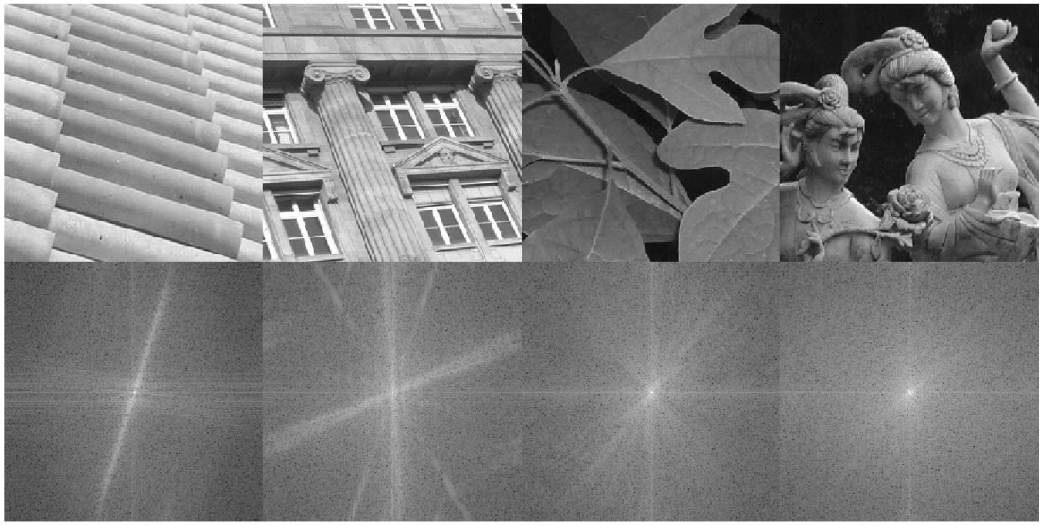


Figure 2.13: Examples for Fourier transforms of 2D images. The lower row shows the Fourier transforms of the images in the upper row. The Fourier transforms are shown as $\log(|1 + F(\boldsymbol{u})|)$, the origin shifted to the centre of the image. **Left:** The shingles are nearly parallel from upper left to lower right. The wave front with the highest frequency is orthogonal to this direction. **Middle left:** The facade shows two dominant directions, which due to the perspective are not mutually orthogonal. The window cornices show edges in two additional directions, which yields the Fourier spectrum with four major directions. **Middle right:** The few stalks of the leaves cause a few only slighly pronounced high frequency directions. **Right:** The smooth contours of the sculptures do not lead to any pronounced high frequencies.

2.6.1 Dirac's Delta Function and the Shah Function

We need two basic functions. They are *generalized functions* since they are defined by a limiting process. We can give their Fourier transform without using the theory of generalized functions. One is the δ-function already mentioned before (see (2.111)), from which we derive the sampling or Shah function $\text{Ш}(x)$.

2.6.1.1 The δ-Function

Dirac's Delta function $\delta(x)$ is the continuous analogion to the unit impulse $\delta(i)$. We may define it by

$$\delta(x) = \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} \tag{2.131}$$

It has the following properties, which we regularly need:

1. The value of the function is 0 or ∞.

$$\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases} \tag{2.132}$$

   The function $a\delta(x - x_0)$ usually is visualized by an arrow at x_0 of height a, as we already did in Fig. (2.11), p. 42 for the Fourier transform of the cosine-function.

2. The area under the function $\delta(x)$ is 1.

$$\int_{-\infty}^{+\infty} \delta(t)\mathrm{d}t = 1 \tag{2.133}$$

Scaling of x by $a \in \mathbb{R}$ reduces the area under the function, hence

$$\delta(ax) = \frac{1}{|a|}\delta(x)\,. \tag{2.134}$$

3. In order to obtain the sampled function value $f(x_0)$ at x_0 we need to convolve $f(x)$ with $\delta(x - x_0)$

$$f(x_0) := f_{x_0}(x) = f(x) * \delta(x - x_0) = f(x - x_0) * \delta(x) \tag{2.135}$$

This is the main motivation to use the δ function. If x_0 is interpreted as a variable, the function $f(x_0)$ has the value of f at x_0 and is zero elsewhere. Observe, sampling $f(x)$ at x_0 in this manner is a mathematically clear concept, however, cannot be realized using any physical system, due to the limited resolution of a sensor.

4. The Fourier transform of $\delta(x)$ is 1

$$\delta(x) \circ\!\!-\!\!\bullet\, 1\,, \tag{2.136}$$

see (2.111), p. 42. Therefore, we also have an alternative definition of the δ-function

$$\delta(x) = \int_{u=-\infty}^{+\infty} \cos(2\pi u x)\mathrm{d}u \tag{2.137}$$

2.6.1.2 The Shah-Function

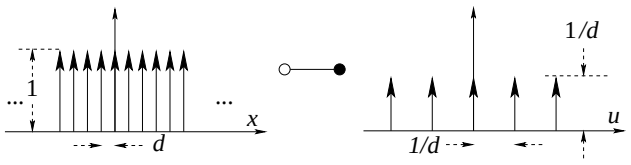


Figure 2.14: The sampling or Shah-function and its Fourier transform. **Left:** The sampling function $\text{Ш}(x/d)/d$ for the sequence $nd, n = ..., -1, 0, +1, ...$, being a train of δ-functions with height 1. **Right:** The Fourier transform of this sampling function again is a sampling function, namely a train of δ-functions at $n/d, n = ..., -1, 0, +1, ...$ and height $1/d$

Die *Shah-function* is an infinite sequence (sometimes called a *train of*) δ-functions:

$$\text{Ш}(x) = \sum_{n=-\infty}^{+\infty} \delta(x - n) \tag{2.138}$$

It is its own Fourier transform

$$\text{Ш}(x) \circ\!\!-\!\!\bullet\, \text{Ш}(u) \tag{2.139}$$

Scaling of x with $1/d$ and f with d in order to preserve the area under the δ-functions, yields

$$\sum_{n=-\infty}^{+\infty} \delta(x - nd) = \frac{1}{d}\text{Ш}\left(\frac{x}{d}\right) \circ\!\!-\!\!\bullet\, \text{Ш}(du) = \frac{1}{d}\sum_{n=-\infty}^{+\infty} \delta\left(u - \frac{n}{d}\right)\,. \tag{2.140}$$

2.6.2 Sampling Theorem and Nyquist Frequency

The sampling theorem states under which condition we can perfectly reconstruct a function from a sampled version of that function. Fig. 2.15 shows the essential steps which are now discussed in more detail.

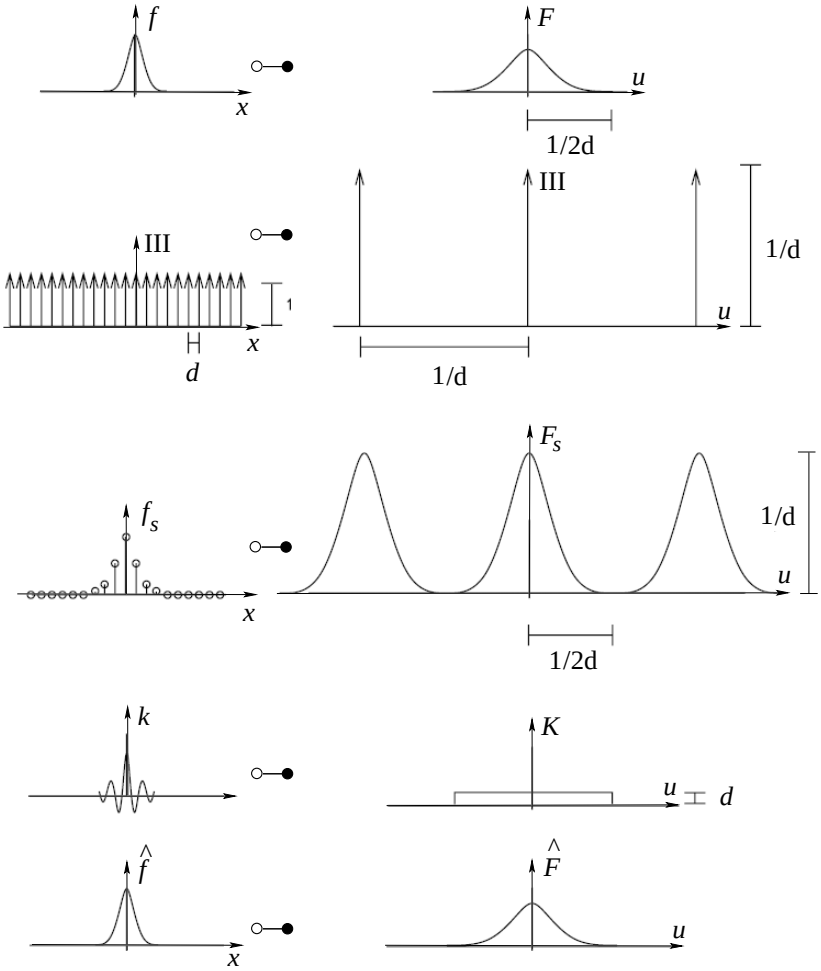


Figure 2.15: Sampling theorem. Left and right: function and its Fourier transform. **1. row:** Given signal $f(x)$ and its Fourier transform $F(u)$. It is assumed to be band limited, i.e., it is zero outside the range $u \in [-1/2d, +1/2d]$. **2. row:** Sampling function with spacing d (here $d < 1$) and its Fourier transform, also being a sampling function. Observe, we assume the sampling to cover the complete range $x \in [-\infty, +\infty]$. **3. row:** Continuous sampled function $f_s(x)$ which is zero for all $x \neq nd$. Its Fourier transform is periodic. The centre peak is proportional to F. **4. row:** Kernel $k(x)$ for recovering the original function from $f_s(x)$. Here it is assumed to be a sinc-function, which has a scaled rectangle-function as Fourier transform. This is used to cut out the central part of the Fourier spectrum of $f_s(x)$. **Last row:** Reconstructed signal $\hat{f}(x)$ with its Fourier transform $\hat{F}(u)$. Here the original function can be recovered without loss since the original function $f(x)$ is band limited, the density of the sampling is high enough, and the complete function $[-\infty, +\infty]$ is sampled . (adapted from www.cs.unm.edu/$\sim$williams/cs530/shannon2.ps)

2.6.2.1 Sampling

Sampling a continuous function with sampling distance d is the transition from the continuous function $f(x)$ to a sequence $(f(id)), i \in \mathbb{Z}$. Given the sequence $(f(id))$ no information on values between integer multiples of d is available.

For analysing the sampling process it is reasonable to define a continuous sampling function $f_s(x)$ which has values $f(id)$ at the positions id and is 0 elsewhere. The generation of the function $f_s(x)$ can be described as multiplication of the function with the Shah function

$$f_s(x) = \sum_{n=-\infty}^{\infty} f(x)\delta(x - nd) = \sum_{k=-\infty}^{\infty} f(id)\delta(x - nd) \tag{2.141}$$

or

$$f_s(x) = f(x) \cdot \frac{1}{d}\text{Ш}(\frac{x}{d}) \,. \tag{2.142}$$

We also can interprete $f_s(x)$ as the Shah function modulated with the function $f(x)$. The function values $f(id)$ are identical to the areas under $f_s(x)$ around id.

Due to the (dual of the) convolution theorem, the Fourier transform of f_s is the convolution of the Fourier transforms of $f(x)$ and $\text{Ш}(x/d)$:

$$F_s(u) = F(u) * \text{Ш}(ud) \tag{2.143}$$

hence

$$F_s(u) = \frac{1}{d} \sum_{n=-\infty}^{+\infty} F(u - \frac{n}{d}) \,. \tag{2.144}$$

The Fourier spectrum of the discretized function $f_s(x)$ is the sum of shifted versions $F(u - n/d)$ of the spectrum $F(u)$. Hence, the spectrum $F_s(u)$ is periodic with period $1/d$.

We therefore only need to analyse the spectrum $F_s(u)$ in the range $[-1/(2d), +1/(2d)]$. If the Fourier spectrum of $f(x)$ is arbitrary, especially if it is long-tailed, summing the shifted versions of $F(u)$ will lead to a function $F_s(u)$ which differs from $F(u)$ in this interval. This is because high frequencies of the spectrum $F(u)$ outside, possibly far outside, the interval will show up as low frequencies, due to the shift. Hence, a reconstruction of $f(x)$ – or, equivalently, of $F(u)$ – from $F_s(u)$ is impossible. This effect is called *aliasing* (from Latin: alias = at another time or place), see the example in Fig. 2.16.

aliasing

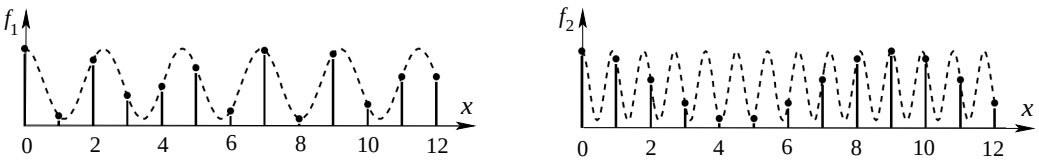


Figure 2.16: Sampling and aliasing. Sampling is done with a distance $\Delta x = d = 1$. **Left:** The sine-function $f_1 = \sin(2\pi u_1 x)$ has frequency $u_1 = 10/9 = 1.111$. The signal appears to be reconstructable from the sampled function values. **Right:** The sine-function $f_2 = \sin(2\pi u_2 x)$ has frequency $u_2 = 10/23 = 0.435$. The reconstruction of the original function appears not possible. We observe a low frequency wave through the sampled points. Hence the high frequencies of the original signal appear somewhere else in the spectrum, namely at higher frequencies. — Following the sampling theorem perfect reconstruction is only possible if $u_0 \leq 1/(2\Delta x)$ where u_0 is the maximum frequency. For $\Delta x = 1$ the maximum frequency must not be larger than $1/\Delta x = 0.5$. Hence, the first signal is reconstructable, the second is not.

It explains Moiree effects, such as appearing when taking digital images of a screen or when observing a wheel in a video rolling backwards. In these cases, the sampling interval is not short enough to catch the high frequencies of the space/time pattern.

2.6.2.2 Reconstruction by Interpolation

We now formalize the reconstruction of a continuous function from a sampled version. This is usually done by *interpolation*. Interpolation can be represented as convolution of

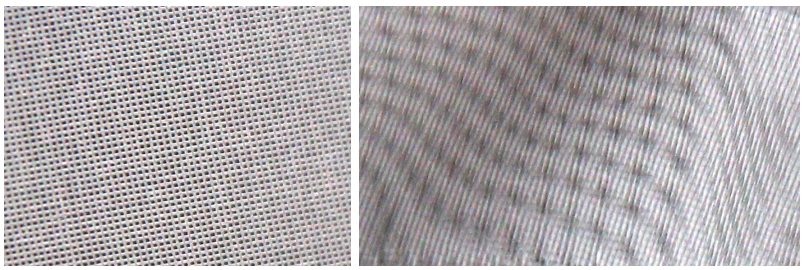


Figure 2.17: Aliasing in 2D. **Left:** Regular pattern of curtain tissue showing the high frequency pattern of the weaving. **Right:** Overlaying two distorted pieces of tissue leads to aliasing effects: a deformed grid pattern with longer wave length. The long range intensity variations result from the non-flatness of the two pieces of tissue. The overlay effect can be interpreted as sampling one of the tissue patterns with the other

$f_s(x)$ with an interpolation kernel $k(x)$. The reconstructed signal thus is

$$\widehat{f}(x) = f_s(x) * h(x)\,. \tag{2.145}$$

The spectrum of this reconstructed signal is

$$\widehat{F}(u) = F_s(u)H(u)\,. \tag{2.146}$$

Classical interpolation kernels are, assuming sampling distance $d = 1$:

1. Nearest neighbourhood interpolation in the spectral domain

$$H_0(u) = \mathrm{rect}(u) = \begin{cases} 1 & \text{for} \quad |u| < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.147}$$

   with its inverse Fourier transform

$$h_0(x) = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{2.148}$$

2. Linear interpolation in the frequency domain with

$$H_1(u) = \Lambda(u) = \begin{cases} 1+u & \text{for} \quad -1 \le u \le 0 \\ 1-u & \text{for} \quad 0 \le u \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.149}$$

   with its inverse Fourier transform

$$h_1(x) = \mathrm{sinc}^2(x) = \frac{\sin^2(x)}{x^2}\,, \tag{2.150}$$

   since $\Lambda(u) = \mathrm{rect}(u) * \mathrm{rect}(u)$.

For a perfect reconstruction of the signal from $f_s(x)$ two conditions need to be fulfilled:

1. There should not occur any aliasing. Therefore, the signal must be band limited, thus have only frequencies below a maximum frequency and the sampling distance must be small enough relative to the shortest wavelength.
2. The interpolation kernel should not change the form of $F_s(u)$ in the interval $[-1/(2d), +1/2d)]$.

Hence, the two mentioned interpolation kernels with sampling distance $d = 1$ are not suited for perfect reconstruction, as the amplitude of already small frequencies may be changed. An interpolation kernel with Fourier spectrum $\mathrm{rect}(U(ud))$ could be used if the maximum frequency of $f(x)$ is $1/(2d)$ or smaller. This leads to the sampling theorem (Whittaker, 1915; Shannon and Weaver, 1949).

Theorem 2.6.8: ***Sampling theorem.*** A continuous function $f(x)$ can only be perfectly reconstructed from the discretized function $f_s(x)$ if

1. the function $f(x)$ is band limited, thus

$$F(u) = 0 \qquad \text{for} \qquad |u| \geq u_0 \tag{2.151}$$

and

2. the sampling interval Δx is small enough, namely

$$u_0 \leq \frac{1}{2\Delta x} \quad \text{or} \quad \Delta x \leq \frac{1}{2u_0} \tag{2.152}$$

The frequency u_0 is called the Nyquist frequency.

The reconstruction can be performed with an interpolation kernel $\text{rect}(u\Delta x)$. This is equivalent to performing the reconstruction as (see (Shannon and Weaver, 1949))

$$f(x) = \sum_{n=-\infty}^{\infty} f_s(n\Delta x)\text{sinc}\left(\frac{x - n\Delta x}{\Delta x}\right) \tag{2.153}$$

If $\Delta x < \frac{1}{2u_0}$ we also can use a kernel $k(x)$ for which $H(u) = 1$ for $|u| < u_0$. Obviously, we would need infinitely many samples for a perfect reconstruction. Therefore good approximations of the sinc-function are used.

Proof: We have

$$F(u) = K(u)F_s(u) \quad \text{with} \quad K(u) = \Delta x\,\text{rect}(u\Delta x)\,. \tag{2.154}$$

The reason for the factor Δx in the spectrum $K(u)$ of the kernel is: The spectrum $F_s(u)$ of the sampled function is is a sum of $F(u - n/\Delta x)$ divided by the sampling distance, see (2.144). Taking the inverse Fourier transformation, we now obtain

$$f(x) = k(x) * f_s(x) \quad \text{with} \quad k(x) = \text{sinc}\left(\frac{x}{\Delta x}\right)\,. \tag{2.155}$$

This reads as

$$f(x) = \int_{y=-\infty}^{\infty} \text{sinc}\left(\frac{x-y}{\Delta x}\right) f_s(y)\text{d}y = \sum_{i=-\infty}^{\infty} \text{sinc}\left(\frac{x - i\Delta x}{\Delta x}\right) f_s(i\Delta x) \tag{2.156}$$

◇

The sampling theorem is relevant when geometrically transforming images especially if they are reduced by large factors. Then we need to perform a low pass filtering before the reduction in order to suppress frequencies outside the Nyquist limit and avoid aliasing effects.

2.7 Other Image Transforms

2.7.1 Wavelets

Representing a signal $f(i)$ as a sum of sine and cosine waves, e.g., as $f(i) = 1/n \sum_{k=0}^{n-1} F_k \exp(\text{i}2\pi ik/n)$ does not allow to analyse the frequency content of a signal locally, since the basis functions refer to the complete range, e.g., $(0, ..., n-1)$, of the signal, e.g.,

$$F(k) = \sum_{i=0}^{n-1} f(i)\exp(-\text{i}2\pi ik/n)\,. \tag{2.157}$$

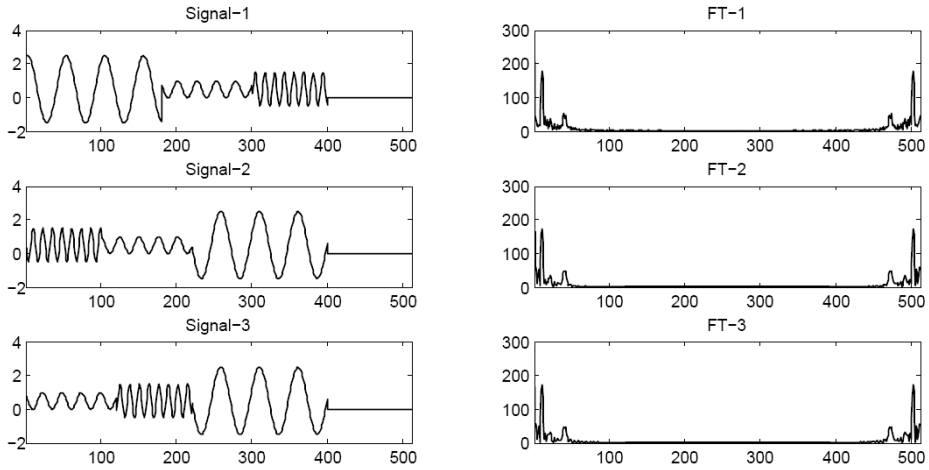


Figure 2.18: Three signals with three permuted sine-waves having mean 0.5, amplitudes $[2.0, 0.5, 1.0]$ and wavelengths $[51.2, 25.6, 12.8]$, covering the range 1...400. The rest of the signal in the range 401...512 is zero. Their amplitude spectrum of all signals is nearly identical. It has three peaks at the frequencies approximately $[51, 26, 13]$, with a height representative for the amplitude

Example 2.7.16: ***Inhomogeneous 1D signals.*** Take as an example a signal consisting of a sequence of sine-waves of different amplitude and frequency followed by a constant signal, see Fig. 2.18 Their amplitude spectrum is – practically – the same. Taking the three peaks at $[51, 26, 13]$, they do not tell where the according waves appear within the signal.

We would rather have a description of the local properties of the signal, ideally – for the first signal – as show in in Fig. 2.19. This can be achieved by convolution of the original signal with windowed sin/cosine waves, which focusses on a limited range in the spatial domain, thus only taking into account a certain neighbourhood of $f(x)$ at each position x, e.g., only two wavelengths. ◇

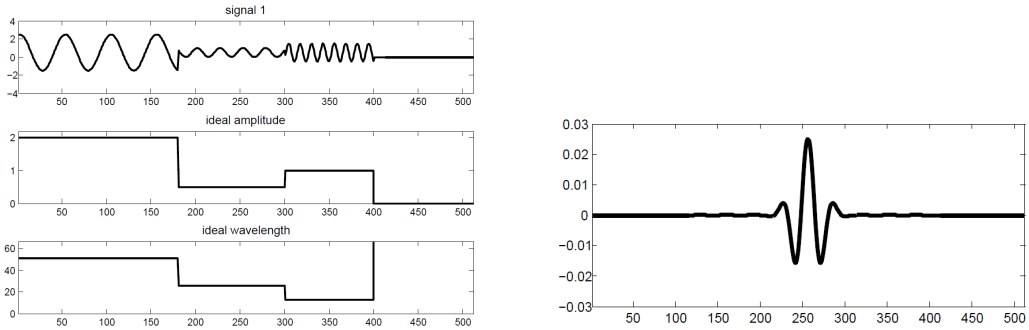


Figure 2.19: Non-homogeneous signal. **Top left:** Sequence of sine waves with wavelengths 51.2, 25.6 ans 12.8. **Mid and lower left** Ideal (desired) response ideal response for amplitude and wavelength. **Lower right:** Real (cosine) part of the Gabor filter with wavelength 32 and position 256

When using as window function $w(i)$ a Gaussian then we arrive at what are called Gabor filters,

$$g_k(i) = w(i) \exp(-\mathrm{i}2\pi ik/n)\,, \tag{2.158}$$

which provide an amplitude/frequency response $f_k(i) = f(i) * g_k(i)$ of the original signal localized in space and frequency. Figure 2.19 shows the real part of such a Gabor filter, which intuitively represents a frequency (wavelength 32, frequency $512/32 = 16$) and a position ($i = 256$).

Unfortunately, generally there is no way to reconstruct the original signal $f(i)$ from a set of filter responses $f_k(i), k = 0, ..., K; i = 0, ..., I$.

This can be achieved by the concept of what is called a *wavelet transformation*. It starts from a basic filter, a small wave, called a *mother wavelet* $\psi(x)$, from which scaled and shifted versions are derived via

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \quad \text{with} \quad j, k, \in \mathbb{Z} \tag{2.159}$$

The scaling is performed in steps of a factor 2, resulting in a compressed signal. The amplitude here is chosen, such that the integral of $\psi^2_{j,k}(x)$ is $\int \psi^2(x)\mathrm{d}x$ independent on (j, k).

The wavelets are similar to the Gabor filter in Fig. 2.19 lower right. The wavelet transform essentially consist of applying the wavelets to a signal leading to a set of filtered signals

$$\mathcal{W}[f] : \; f \mapsto c \quad c_{j,k}(x) = \psi_{j,k}(x) * f(x)\,, \tag{2.160}$$

tells the frequency and the position in the original signal. The concept of wavelets requires that the original signal $f(x)$ can be recovered from the transformed signals $c_{j,k}(x)$ via

$$f(x) = \sum_{j,k} c_{j,k}(x)\psi_{j,k}(x)\,. \tag{2.161}$$

There are various ways how such wavelets can be defined, see Daubechies (1992). We will discuss the most basic wavelet, the Haar wavelet $h_k(i)$, see Haar (1910). It is useful for deriving a rich representation of a signal, and therefore used in texture analysis and object recognition.

Since Haar wavelets show discontinuities, of the above mentioned Gabor filters $g_{j,k}$ are used, which also can be defined starting from a basic filter $g(x)$ by scaling and shifting according to (2.159) and thus allow a local frequency analysis of the signal. Such Gabor wavelets show a similar behaviour as cells in the visual cortex, and have regularly been used for texture analysis, why we discuss them more in detail. However, they do not allow a reconstruction of $f(x)$ from its transformed elements $f(x) * g_{j,k}(x)$.

Finally, we discuss what are called *steerable filters* which eases the design and increases the efficiency of oriented filters. Steerable filters allow to realize rotated filters as a *weighted sum* of basic filters. They require that the basic filters have have specific properties and show that partial derivatives of Gaussian filters belong to this class, which are regularly used for edge and line extraction.

2.7.2 Haar Wavelets

The most simple mother wavelet is the function, proposed by Haar (1910)

$$\psi(x) = \begin{cases} 1, \text{ if } x \in [0, 1/2) \\ -1, \text{ if } x \in [1/2, 1) \\ 0, \text{ else.} \end{cases} \quad \text{with} \quad \int \psi(x)\mathrm{d}x = 0 \quad \text{and} \quad \int \psi^2(x)\mathrm{d}x = 1\,. \tag{2.162}$$

It is shown in Fig. 2.20, upper left. If we do not channge the amplitude, we obtain the functions $\psi_{j,k}$ for $j = 0, ..., 2; k = 0, ..., 2^j - 1$. These basis functions can approximate functions with mean zero in the interval $[0, 1]$ using (2.160).

Assuming discrete functions $\boldsymbol{f} = [f_i]$ in the range $i \in [0, 2^n - 2]$ the basis functions in a natural manner are related to the discrete Haar transform

$$\boldsymbol{c} = \boldsymbol{H}\boldsymbol{f} \tag{2.163}$$

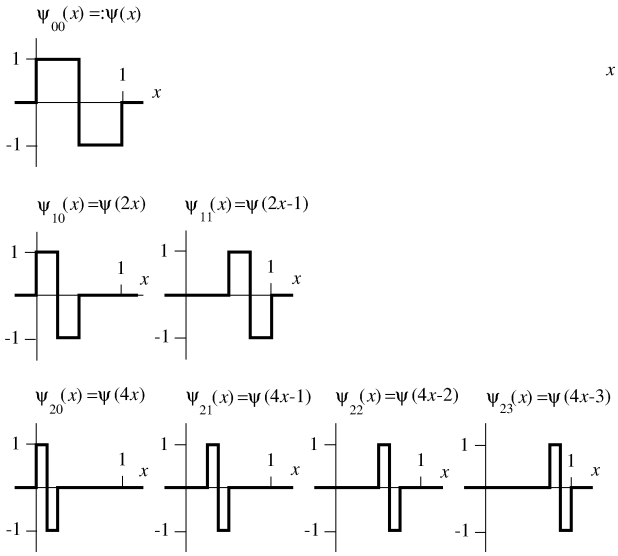


Figure 2.20: Haar wavelets $\psi_{jk}, j = 0, ..., 2; k = 0, ..., 2^j - 1$, from Weisstein (2023)

with the un-normalized matrices

$$H^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.164}$$

$$H^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{2.165}$$

$$H^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{2.166}$$

Observe, the first row of the matrices is meant to take the mean, while the other rows are shifted differentiation filters $\mathbf{1} * [1\ , -1]$. The matrices $H^{(2^j)}$ are orthogonal but not orthonormal, since the factor $2^{j/2}$ in (2.159) is not applied. The normalized matrices read

as

$$\overline{H}^{(1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.167}$$

$$\overline{H}^{(2)} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \tag{2.168}$$

$$\overline{H}^{(3)} = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \tag{2.169}$$

such that for the orthonormal matrices $\overline{H}^{(2^j)}$ holds

$$\overline{H}^j \overline{H}^{j)\mathsf{T}} = I_{2^j} \,. \tag{2.170}$$

There is a simple recursive relation for the general non-normalized Haar matrices j

$$H^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.171}$$

$$H^{j+1} = \begin{bmatrix} \overline{H}^{(j)} \otimes [1 \quad 1] \\ I_{2^j} \otimes [1 \; -1] \end{bmatrix} \quad \text{for} \quad j > 1,, \tag{2.172}$$

modifying the rule given by Steeb et al. (2003), which holds for the normalized Haar matrices

$$\overline{H}^{(1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.173}$$

$$\overline{H}^{j+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} \overline{H}^{(j)} \otimes [1 \quad 1] \\ I_{2^j} \otimes [1 \; -1] \end{bmatrix} \quad \text{for} \quad j > 1 \,. \tag{2.174}$$

Similar to using only a subset of waves for approximating a signal via a Fourier transform in Fig. 2.9, we may approximate a signal using only the a subset of Haarwavelets, see Fig. 2.21.

As already mentioned above, Haar wavelets have the disadvantage of leading to discontinuous approximations. Therefore, we address gabor filters in the next section.

The Haar transformation can be applied to 2D signals, similar to the Fourier transformation in (2.103). For simplicity, let the 2D signal be the square $2^j \times 2^j$ matrix A then its (non-normalized) Haar transform is given by (omitting an indicator for the type of transform)

$$\mathcal{H}(g): \quad B = HAH^{\mathsf{T}} \,. \tag{2.175}$$

We can interprete this expression taking the rows of the Haar matrix

$$H = \begin{bmatrix} \boldsymbol{h}_1^{\mathsf{T}} \\ \ldots \\ \boldsymbol{h}_m^{\mathsf{T}} \\ \ldots \\ \boldsymbol{h}_{2^j}^{\mathsf{T}} \end{bmatrix} \tag{2.176}$$

and expressing elements B_{mn}

$$B_{mn} = \boldsymbol{h}_m^{\mathsf{T}} A \boldsymbol{h}_n = \mathrm{tr}\left(\boldsymbol{h}_n \boldsymbol{h}_m^{\mathsf{T}} \, A\right) = \mathrm{tr}\left((\boldsymbol{h}_m \boldsymbol{h}_n^{\mathsf{T}})^{\mathsf{T}} \, A\right) \,. \tag{2.177}$$

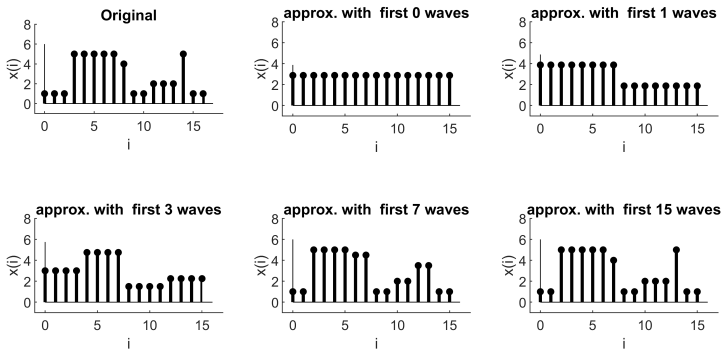


Figure 2.21: Approximating an original signal using a subset of the Haar wavelets, namely of index up to 1, 2, 4, 8, and 16, the last option yielding the original signal, , see the difference to the result in Fig. 2.9, where the mean signal is not shown

With the individual 2D Haar wavelets

$$H_{mn} = \boldsymbol{h}_m \boldsymbol{h}_n^{\mathsf{T}} \tag{2.178}$$

this can be written as

$$B_{mn} = \sum_{k=1,\ldots,2^j, l=1,\ldots,2^j} (H_{mn})_{kl} B_{kl} = \mathbf{1}^{\mathsf{T}} (H_{kl} \circ B) \mathbf{1} \tag{2.179}$$

expressing the sum of all elements of B weighted with the (kl)-matrix H_{kl} of the 2D Haar transformation (the symbol $\circ$ denotes elementwise multiplication).

The complete set of the non-normalized Haar matrices for the cases $j = 1$ and $j = 3$ are shown in Fig. 2.22.

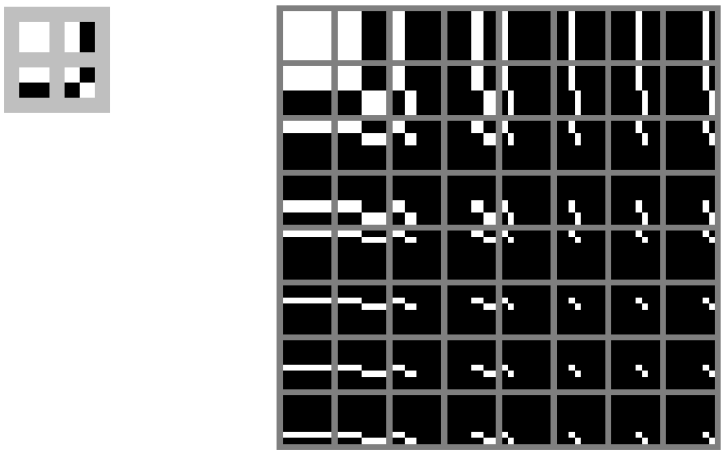


Figure 2.22: 2D Haar wavelets. **Left:** Case $j = 1$ with four 2×2 wavelets. **Right:** Case $j = 3$ with 64 8×8 wavelets.

We start with the case $j = 1$.

- The set consists of 2×2 2D Haar wavelets of size 2×2.
- The upper left wavelet is responsible for the mean of the signal.
- The off diagonal wavelets – up to a scaling – represent the elementary discrete differentiation filters $\Delta/\Delta x$ and $\Delta/\ \Delta y$ in column and row direction.

- The lower right wavelet represents the torsion filter $\Delta^2/(\Delta x \Delta y)$.

All filters have the same size; thus the differentiation filters contain a smoothing part, e.g., here the smoothing in row direction:

$$\frac{\Delta}{\Delta x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} * [1 \;\; 1] = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \tag{2.180}$$

For the case $j = 3$ we obtain 8×8 2D wavelets of size 8×8. Again the upper left wavelet is resopnsible for the mean. The other wavelets are either differentiaon filters or torsion filters, possibly differently sized in x- and y-direction.

This suggest to interpret the Haar transformation as correlation of A with each wavelet:

$$B_{mn} = \mathsf{H}_{mn} \circledast \mathsf{B} \quad \text{with} \quad m = 1, ..., 2^j, n = 1, ..., 2^j \tag{2.181}$$

Since A and H_{mn} have the same size, the result is a scalar, the value B_{mn}.

Interpreting the wavelets as filters, allows to perform a type of scale analysis, useing the unshifted wavelets ($k = 0$) as a filter bank, see Viola and Jones (2001), where in addition also line type features are applied.

2.7.3 Gabor Wavelets

Gabor Filters in 1D. Gabor filters allow to analyse the local frequency content of a signal. They need to fulfill two conflicting requirements: They need to focus on a certain position and on a certain frequency. We will see that the position and the frequency of a signal cannot be determined with arbitrary accuracy, similar to the uncertainty principle in quantum physics, which states that the position and the momentum of a particle cannot be measured simultaneously with arbitrary precision.

For a local frequency analysis let us take a certain frequency u_0 and, instead of using the complete sine or cosine wave, or the complete complex wave $\exp(\mathrm{i}2\pi x u_0)$, let us take a windowed version of it, by multiplying $\exp(\mathrm{i}2\pi x u_0)$ with a window function $w(x)$, which we – following Gabor – assume to be a Gaussian, positioned at $x = 0$ and having a certain width s_0. Hence, we obtain a filter of the following form:

$$G(x|u_0; s_0) = \frac{1}{\sqrt{2\pi s_0^2}} \exp\left(-\frac{x^2}{2s_0^2}\right) \exp(\mathrm{i}2\pi x u_0)\,, \tag{2.182}$$

or equivalently a windowed even (cosine) or odd (sine) filter

$$G^e(x|u_0; s_0) = \frac{1}{\sqrt{2\pi s_0^2}} \exp\left(-\frac{x^2}{2s_0^2}\right) \cos(2\pi x u_0) \tag{2.183}$$

$$G^o(x|u_0; s_0) = \frac{1}{\sqrt{2\pi s_0^2}} \exp\left(-\frac{x^2}{2s_0^2}\right) \sin(2\pi x u_0)\,. \tag{2.184}$$

Since the Fourier transform of the convolution of two signals is the product of the Fourier transforms, and this also holds for the inverse Fourier transform, we obtain the Fourier transform of the Gabor filter as the convolution of the Gaussian window, which again is a Gaussian function and the Fourier transform of the pure exponential which is a sum of two delta functions:

$$\mathcal{F}(G(x)) = \mathcal{F}(h(x)) * \mathcal{F}(\exp(\mathrm{i}2\pi x u_0))\,. \tag{2.185}$$

Due to (2.110) this is $\exp(-2\pi^2 s_0^2 u^2) * \delta(u - u_0)$, hence a (non-normalized) Gaussian, which has centre u_0:

$$\mathcal{F}(G(x)) = \exp\left(-2\pi^2 s_0^2 (u - u_0)^2\right)\,. \tag{2.186}$$

As we can see from Fig. 2.23, the filter is concentrated in both, the spatial domain and the frequency domain. The concentration depends on the chosen width s_0 of the window function. The concentration can be measured by the normalized second central moments,

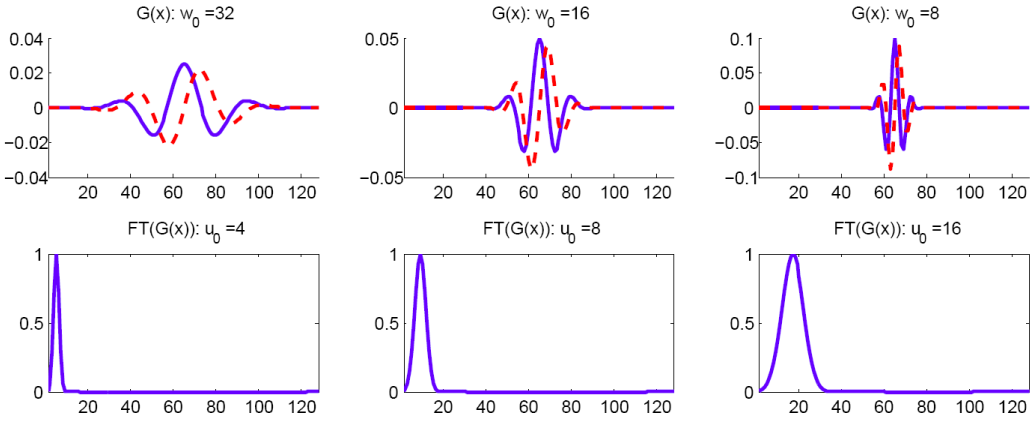


Figure 2.23: Three Gabor filters with length $N = 128$, and different wavelengths w_0 and frequencies $u_0 = N/w_0$, centred at $N/2$. The width s_0 has been adapted to the wavelength in order to capture the same number of waves, namely $s_0 = w_0/2$. **Left:** $w_0 = 32$, $u_0 = 4$. **Middle:** $w_0 = 16$, $u_0 = 8$. **Right:** $w_0 = 8$, $u_0 = 16$. **Top row:** Even (blue) and odd (red dashed) Gabor filters. **Bottom row:** Fourier transform of the filters: Gaussian with peak at frequency u_0. The product of the width σ_x of the discrete signal in the spatial domain and the width σ_u in the discrete frequency domain is independent of s_0 and equal to $N^2/(16\pi^2)$

i.e., the variances of the squared functions $|G(x)|^2$ and $|\mathcal{F}(G(x))|^2$, in order to eliminate the negative and imaginary parts. The variances therefore are

$$\sigma_x^2(s_0) = \frac{\int_{x=-\infty}^{\infty} x^2 |G(x)|^2 \mathrm{d}x}{\int_{x=-\infty}^{\infty} |G(x)|^2 \mathrm{d}x} = \frac{1}{2} s_0^2 \,, \tag{2.187}$$

and

$$\sigma_u^2(s_0) = \frac{\int_{x=-\infty}^{\infty} (u-u_0)^2 |\mathcal{F}(G(x))|^2 \mathrm{d}u}{\int_{x=-\infty}^{\infty} |\mathcal{F}(G(x))|^2 \mathrm{d}u} = \frac{1}{8\pi^2 s_0^2} \,. \tag{2.188}$$

Hence the product of both variances is

$$\sigma_x^2(s_0)\, \sigma_u^2(s_0) = \frac{1}{16\pi^2} \,, \tag{2.189}$$

independent of the width s_0. The equation states, that refining the positioning accuracy, by diminishing s_0, leads to a less well defined frequency description, a larger σ_u, and vice versa. Hence, we cannot determine the frequency content of a signal at a well-defined position with arbitrary accuracy. This is the equivalent to Heisenberg's uncertainty principle. If the window function is not chosen to be a Gaussian, then it can be proven, that the product is larger than $1/(16\pi^2)$, thus generally we have (see Folland and Sitaram (1997))

$$\sigma_x^2\, \sigma_u^2 \geq \frac{1}{16\pi^2} \,. \tag{2.190}$$

Gabor Filters in 2D. The concept of windowed filters can easily be transferred to 2D. here, we take a basis function sine and cosine waved with a frequency of w_0 in a certain direction ϕ, see Fig. 2.24. With the direction vector $\boldsymbol{n} = [n_x, n_y] = [\cos\phi, \sin\phi]^\mathsf{T}$ we start from

$$\cos(2\pi \boldsymbol{n}^\mathsf{T} \boldsymbol{x} w_0) = \cos(2\pi(x\cos\phi + y\sin\phi)w_0) \,. \tag{2.191}$$

This is an infinite wall roof pattern with Fourier transform

$$\mathcal{F}(\cos(2\pi \boldsymbol{n}^\mathsf{T} \boldsymbol{x} w_0)) = \frac{1}{2}[\delta(\boldsymbol{u} - \boldsymbol{u}_0) + \delta(\boldsymbol{u} + \boldsymbol{u}_0)] \quad \text{with} \quad . \tag{2.192}$$

with the vector

$$\boldsymbol{u}_0 = \boldsymbol{n} f_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} w_0 \,. \tag{2.193}$$

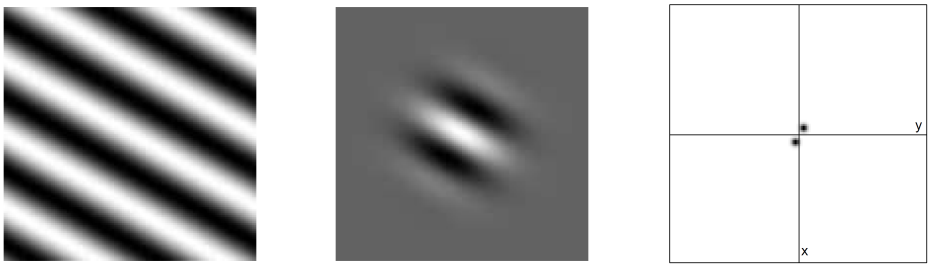


Figure 2.24: Gabor filter: **Left:** Wave with given wavelength and direction. **Mid:** its localized version, a 2D-Gabor filter, **Right:** The absolute value of the 2D-Fouriertransform, also localized

In order to focus the analysis onto a certain position (x, y), we again apply a Gaussian window, especially a isotropic Gaussian with width s_0 and achieve a 2D Gabor filter

$$G(x, y|u_0, v_0, s_0) = \frac{1}{2\pi s_0^2} \exp\left(-\frac{||\boldsymbol{x}||^2}{2s_0^2}\right) \exp\left(\mathrm{i}2\pi[u_0 x + v_0 y]\right) , \tag{2.194}$$

or more compactly

$$G(\boldsymbol{x}|\boldsymbol{u}_0, s_0) = \frac{1}{2\pi s_0^2} \exp\left(-\frac{||\boldsymbol{x}||^2}{2s_0^2}\right) \exp\left(\mathrm{i}2\pi \boldsymbol{u}_0^\mathsf{T} \boldsymbol{x}\right) , \tag{2.195}$$

which again has an even (cosine) and an odd (sine) component:

$$G^e(\boldsymbol{x}|\boldsymbol{u}_0, s_0) = \frac{1}{2\pi s_0^2} \exp\left(-\frac{||\boldsymbol{x}||^2}{2s_0^2}\right) \cos\left(2\pi \boldsymbol{u}_0^\mathsf{T} \boldsymbol{x}\right) \tag{2.196}$$

$$G^o(\boldsymbol{x}|\boldsymbol{u}_0, s_0) = \frac{1}{2\pi s_0^2} \exp\left(-\frac{||\boldsymbol{x}||^2}{2s_0}\right) \sin\left(2\pi \boldsymbol{u}_0^\mathsf{T} \boldsymbol{x}\right) . \tag{2.197}$$

The Fourier transform of the Gabor filter again is a non-normalized Gaussian centred at $\boldsymbol{u}_0 = [u_0, v_0]^\mathsf{T}$,

$$\mathcal{F}\left(G(x, y)\right) = \exp\left(-2\pi^2 s_0^2 [(u - u_0)^2 + (v - v_0)^2]\right) . \tag{2.198}$$

As in the one-dimensional case, the filter is located well in both, the spatial and the frequency domain, as can be seen in Fig. 2.24.

Gabor Wavelets. Gabor filters obviously can be designed to cover all frequencies and – in two dimensions all directions – and thus are able to provide a full analysis of the local spectral properties of a signal. As already seen in Fig. 2.23 the different Gabor filters can be derived from what is called a basic *mother wavelet* by dilation and – in two dimensions by rotation. In the following we only refer to two-dimensional images.

Let a Gabor filter $G(\boldsymbol{x}|f, \phi)$ be characterized by its direction ϕ and its frequency f. Starting from $\phi_0 = 0$ and some smallest frequency f_0 and assume the width of the Gaussian window $h(x)$ is half the wavelength, hence $s_0 = 1/(2f_0)$, then we have the Gabor mother wavelet

$$G(\boldsymbol{x}|f_0, \phi_0) = \frac{f_0^2}{2\pi} \exp\left(-2f_0^2 ||\boldsymbol{x}||^2\right) \exp\left(\mathrm{i}2\pi \boldsymbol{x}^\mathsf{T} \boldsymbol{u}_0(f_0, \phi_0)\right) \quad \text{with} \quad \boldsymbol{u}(f, \phi) = f \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \tag{2.199}$$

with its Fourier transform

$$\mathcal{F}\left(G(\boldsymbol{x}|f_0, \phi_0)\right) = \exp\left(-\frac{\pi^2}{2f_0^2} ||\boldsymbol{u} - \boldsymbol{u}_0(f_0, \phi_0)||^2\right) \tag{2.200}$$

If we now want to cover the frequency plane by a set of Gabor filters, we may choose a set of directions ϕ and a set of frequencies f. Since we used an isotropic weight function, each

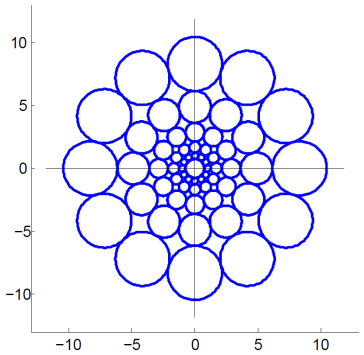


Figure 2.25: The frequency domain can be tesselated with Gabor filters: A fixed directional resolution can be transferred to a fixed ratio of the frequencies, i.e., distances to the origin

filter can be represented by a circle in the frequency plane. Hence, if we want to have a circular band tesselated with circles, then, if we choose the number M_ϕ of directions, the scale change between the frequencies is fixed, as can be seen in Fig. 2.25. There we chose the smallest frequency $f_0 = 1$ and the number of frequencies $M_f = 5$. then we arrive at other Gabor filters by choosing a set of directions ϕ and a set of frequencies f.

2.8 Exercises

1. Show the relation of the real and the complex coefficients B_n, φ_n and B'_n in (2.4) and (2.5). Are there relations between the coefficients B_n? Be specific.
2. Prove the filter kernel for the backward and the forward differences (2.22) and (2.23).
3. Let a discrete signal $(x_i), i = -\infty, ..., \infty$ be smoothed using a moving average of the five elements around it:

$$y_i = \frac{1}{5} \sum_{j=i-2}^{i+2} x_j \quad \text{with} \quad i = -\infty, ..., \infty\,. \tag{2.201}$$

   Show, that this equation can be written as a convolution $y = r * x$. Give the kernel r explicitly.
4. Use (2.73), p. 35 to show that the elements $[x_i]$ are real if the elements $[X_k]$ of its discrete Fourier transform are related by $X_k = X^*_{n-k}$. Use this result and show, that only n real values are necessary for representing the Fourier spectrum of a real signal. *Hint:* Discuss signals with odd and even elements separately.
5. For the reverse signal $\overline{\boldsymbol{x}} = [x_0, x_{n-1}, ..., x_1]$ show that the Fourier transforms of $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}$ are related by $\overline{\boldsymbol{X}} = \boldsymbol{x}^*$.
6. Given is a random n-vector $\underline{\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{xx})$ with the cyclical covariance matrix $\Sigma_{xx} = Z(\boldsymbol{c})$. Its Fourier transform is given by $\underline{\boldsymbol{X}} = F\underline{\boldsymbol{x}}$ or $\underline{X}_k = \boldsymbol{f}_k^\mathsf{T}\underline{\boldsymbol{x}}$ with $\boldsymbol{f}_k^\mathsf{T} = [1, w^{-k}, w^{-2k}, ..., w^{-(n-1)k}]$ with the unit root $w = \exp(2\pi\mathrm{i}/n)$. Derive the 2×2 covariance matrix $\Sigma_{U_kU_k} := \mathbb{D}(\underline{\boldsymbol{U}}_k)$ of the vector $\underline{\boldsymbol{U}}_k = [\Re(\underline{\boldsymbol{X}}_k), \Im(\underline{\boldsymbol{X}}_k)]$ and the variance of $\underline{X}_k$. *Hint:* Use the two vectors $\boldsymbol{a}_k = \Re(\boldsymbol{f}_k)$ and $\boldsymbol{b}_k = \Im(\boldsymbol{f}_k)$.
   (a) Under which conditions is $b_k = \mathbf{0}$? Give the 2×2-covariance matrix $\Sigma_{U_kU_k}$ for these cases.
   (b) Use the decomposition $Z(\boldsymbol{c}) = \sum_{i=0}^{n-1} Z_i = \sum_{i=0}^{n-1} c_i Z(\boldsymbol{e}_i)$, where $\boldsymbol{e}_i$ is the i-th unit vector and derive explicit expressions for the bilinear forms $\boldsymbol{a}_k^\mathsf{T} Z_i \boldsymbol{a}_k$, $\boldsymbol{a}_k^\mathsf{T} Z_i \boldsymbol{a}_k$, as a function of the unit roots w for the case that $\boldsymbol{b}_k \neq \mathbf{0}$. *Hint:* Use $\Re(a + \mathrm{i}b) = (\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x})/2$ and $\Im(a + \mathrm{i}b) = (\mathrm{e}^{\mathrm{i}x} - \mathrm{e}^{-\mathrm{i}x})/2$. Simplify notation by setting $\sum_i := \sum_{i=0}^{n-1}$.

(c) Give the 2×2 covariance matrices of $\underline{\boldsymbol{U}}_k$ for the cases that $\boldsymbol{b}_k \neq \boldsymbol{0}$.

(d) Which distribution do the Fourier coefficients $\underline{\boldsymbol{U}}_k$ have.

(e) Which distribution do the coefficients of the empirical power spectrum have? *Hint:* Why is it necessary here to take into account whether $\boldsymbol{b}_k$ is the zero vector?

(f) The Fourier coefficients are linear functions of the signal, hence, their real an imaginary part are normally distributed.

(g) Since the coefficients of the power spectrum are the sum of two squared normally distributed random variables.

In case of real $\boldsymbol{f}_k$ the imaginary part has variance zero. Therefore, the coefficients $\underline{C}_k, k = 0, n/2$ are the square of a normally distributed random variable, namely $\underline{C}_k = \underline{X}_k^2$ with $\underline{x}_k \sim \mathcal{N}(0, \sigma_{X_k}^2)$. Hence, we have the chi-square distributed variables

$$\frac{\underline{C}_k}{\sigma_{X_k}^2} \sim \chi_1^2 \quad \text{for} \quad k \in \{0, n/2\}\,. \tag{2.202}$$

In all other cases the real and the imaginary part of $\underline{X}_k$ have the same variance, namely

$$\mathbb{V}(\Re(\underline{X}_k)) = \mathbb{V}(\Im(\underline{X}_k))) = \frac{\sigma_{X_k}^2}{2}\,. \tag{2.203}$$

Therefore we have the chi-square distributed variables

$$\frac{\underline{C}_k}{\sigma_{X_k}^2/2} \sim \chi_2^2 \quad \text{for} \quad k \notin \{0, n/2\}\,. \tag{2.204}$$

3 Logistic Regression

Logistic Regression is a basic tool for classification. The first part of the note, taken from a lecture, provides the basics for two class classification, generalizes to multiple classes and nonlinear decision boundaries, and provides some examples. The second part, more of conceptual nature, had been an internal basis for the lecture, highlights the structure of the decision boundaries, and focusses on the structure of the Hessian matrix and its rank.

3.1 Preface

This notes consists of two parts. The first part is a section from a lecture on pattern recognition from 2009. It provides the basics of logistic regression, and generalizes to multiple classes and to nonlinear decision boundaries. The second part (2010/2022) is more of conceptual nature and served as an internal basis for the first part. It summarizes the note by Jordan (1995) and parts of the paper on the relation between Bayesian networks and logistic regression by Roos et al. (2005) and especially focuses on rank and the use of the Hessian during parameter learning, which is not discussed in Bishop (2006, Sect. 4.3.3).

3.2 Lecture on Logistic Regression

3.2.1 Motivation and Problem

We discuss a method for classification that has the following favourable properties:

- The number of classes can be two or more.
- The number of features can be arbitrary.
- The class boundaries are linear. By transforming the features, we may realize non-linear class boundaries.
- The number of parameters for the classifier increases linearly with the number of classes
- There is a more efficient algorithm for Determination of the parameters from training data.

- The classifier returns for each to be classified object the a posteriori probability of class membership.
- The underlying model is discriminative, i.e., the procedure serves to separate the classes.

We start with the binary classification, motivate the structure of the classification model, illustrate the functions involved and show that determining the parameters from training data leads to a convex optimization problem. Then we formulate the problem for more than two classes and show that non-linear class boundaries can also be realized.

3.2.2 The logistic function and the basic model of logistic regression

The so-called logistic function determines the name

$$\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{3.1}$$

with remarkable properties:

$$\begin{aligned}
\sigma(-x) &= 1-\sigma(x) && (3.2)\\
\sigma'(x) &= \sigma(x)(1-\sigma(x)) = \sigma(x)\sigma(-x) && (3.3)\\
2\sigma(x) &= 1+\tanh\left(\frac{x}{2}\right) && (3.4)\\
\lim_{t\to 0}\sigma\left(\frac{x}{t}\right) &= H(x) && (3.5)
\end{aligned}$$

Its derivative $\sigma'(x)$ looks similar to the Gaussian function, but it does not fall so quickly. One can therefore also use $\sigma(x)$ as a model for a soft step function. From $\sigma(x/t)$ in the limit one obtains a step function, also called *Heaviside function*, (3.5).

We start with the binary maximum a posteriori classification of an object based on a scalar feature. Given the class $\omega_i, i = 1.2$, we first assume the feature is normally distributed with the class independent variance σ but different mean $\mu_i \doteq \mu_{x|\omega_i}, i = 1.2$. Then we get the posterior distribution

$$\begin{aligned}
P(\omega_1|x) &= \frac{p(x|\omega_1)P(\omega_1)}{p(x|\omega_1)P(\omega_1)+p(x|\omega_2)P(\omega_2)} && (3.6)\\
&= \frac{1}{1+\dfrac{p(x|\omega_2)P(\omega_2)}{p(x|\omega_1)P(\omega_1)}} && (3.7)\\
&= \frac{1}{1+\exp\left(\ln p(x|\omega_2)+\ln P(\omega_2)-\ln p(x|\omega_1)-\ln P(\omega_1)\right)}\\
&= \frac{1}{1+\exp\left(\ln G(x|\mu_2,\sigma^2)-\ln G(x|\mu_1,\sigma^2)+\ lnP(\omega_2)-P(\omega_1)\right)}
\end{aligned}$$

and with

$$\begin{aligned}
\ln G(x|\mu,\sigma^2) &= \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)\right) && (3.8)\\
&= \ln\frac{1}{\sqrt{2\pi\sigma^2}}+\frac{-x^2}{2\sigma^2}+\frac{2\mu x}{2\sigma^2}+\frac{-\mu^2}{2\sigma^2} && (3.9)\\
&= \underbrace{\ln\frac{1}{\sqrt{2\pi\sigma^2}}+\frac{-x^2}{2\sigma^2}}_{\text{independent of }\omega}+\frac{\mu(2x-\mu)}{2\sigma^2} && (3.10)
\end{aligned}$$

we receive

$$P(\omega_1|x) = \frac{1}{1+\exp\left(\frac{\mu_2(2x-\mu_2)}{2\sigma^2}-\frac{\mu_1(2x-\mu_1)}{2\sigma^2}+\ln P(\omega_2)-\ln P(\omega_1)\right)}$$

Obviously, the exponent in the denominator is linear in x. Therefore, we can write

$$P(\omega_1|x) = \frac{1}{1+\exp(w_0+w_1x)} \tag{3.11}$$

with

$$w_0 = \frac{\mu_1^2-\mu_2^2}{2\sigma^2} + \ln P(\omega_2) - \ln P(\omega_1) \tag{3.12}$$

$$w_1 = \frac{\mu_2-\mu_1}{2\sigma^2} \tag{3.13}$$

Correspondingly we get

$$P(\omega_2|x) = 1 - P(\omega_1|x) = \frac{\exp(w_0+w_1x)}{1+\exp(w_0+w_1x)} \tag{3.14}$$

Now let us look at the logarithm of the ratios of the posterior probabilities

$$\boxed{\ln\frac{P(\omega_2|x)}{P(\omega_1|x)} = w_0 + w_1x} \tag{3.15}$$

to see that it is a linear function in x.

We now assume that this model is used for classification, even if the distribution of features is not normal: The logarithm of the two a posteriori probabilities is given as a linear function of the feature modelled.

Remark: The logistic function (3.1) gives the model (3.11) because of (3.15) the name: logistic regression. The term regression is misleading, since logistic regression as such is a classification method, though internally using a regression method. ◇

The classification model can thus be formulated as follows: for a given feature x choose the class ω_i with maximum a posteriori probability:

$$\hat{\omega} = \text{argmax}_i P(\omega_i|x,\mathbf{w}) \tag{3.16}$$

with

$$P(\omega_i|x,\mathbf{w}_i) = \frac{\exp\left(\mathbf{w}_i^\mathsf{T}\mathbf{x}\right)}{\sum_i \exp\left(\mathbf{w}_i^\mathsf{T}\mathbf{x}\right)} = \frac{\exp\left(w_{0i}+w_{1i}x\right)}{\sum_i \exp\left(w_{0i}+w_{1i}x\right)} \tag{3.17}$$

with the parameters and the homogeneous feature vector

$$\mathbf{w}_1 = \begin{bmatrix} w_{01} \\ w_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{w} := \mathbf{w}_2 = \begin{bmatrix} w_{02} \\ w_{12} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \tag{3.18}$$

The formulation (3.17) is symmetric in both classes. However, we have fixed the parameters for the first class and combined the parameters for the second class in the vector $\mathbf{w}$. This does not change the decision boundaries, see the argument before Eq. (3.56).

3.2.3 Learning the parameters

Now suppose we have N training examples available. These are couples $(\omega_n, x_n)_n, n = 1, ..., N$, i.e. N objects with their properties x_n and the classes ω_n. The task is to determine optimal parameters $\mathbf{w}$. To do this, we collect the pairs in the two vectors

$$\underset{N\times 1}{\boldsymbol{\omega}} = \begin{bmatrix} \omega_1 \\ \dots \\ \omega_n \\ \dots \\ \omega_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{bmatrix} \qquad \underset{N\times 2}{X} = \begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \dots \\ \mathbf{x}_n^\mathsf{T} \\ \dots \\ \mathbf{x}_N^\mathsf{T} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{3.19}$$

such that the first N_1 pairs are grouped in $\boldsymbol{\omega}_1$ and $\boldsymbol{X}_1$ and the other N_2 pairs in $\boldsymbol{\omega}_1$ and $\boldsymbol{X}_1$. Apparently $N = N_1 + N_2$ applies.

The parameters are optimal if, given data $\boldsymbol{X}$ and parameters $\mathbf{w}$, the overall probability for the classes $\boldsymbol{\omega}$ is greatest. So we want $\mathbf{w}$ from

$$\hat{\mathbf{w}} = \text{argmax}_{\mathbf{w}} P(\boldsymbol{\omega}|\boldsymbol{X}, \mathbf{w}) \tag{3.20}$$

with

$$P(\boldsymbol{\omega}|\boldsymbol{X}, \mathbf{w}) = \prod_n P(\omega_n|\mathbf{x}_n, \mathbf{w}) \tag{3.21}$$

Instead of $P(\boldsymbol{\omega}|\boldsymbol{X}, \mathbf{w})$ we can also maximize the logarithm:

$$\begin{aligned}
Q(\mathbf{w}) &= \ln P(\boldsymbol{\omega}|\boldsymbol{X}, \mathbf{w}) && (3.22)\\
&= \sum_n \ln P(\omega_n|\mathbf{x}_n, \mathbf{w}) && (3.23)\\
&= \sum_{n=1}^{N_1} \ln P(\omega_1|\mathbf{x}_n, \mathbf{w}) + \sum_{n=N_1+1}^{N} \ln P(\omega_2|\mathbf{x}_n, \mathbf{w}) && (3.24)\\
&= \sum_{n=1}^{N_1} \ln \frac{1}{1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x})} + \sum_{n=N_1+1}^{N} \ln \frac{\exp(\mathbf{w}^\mathsf{T}\mathbf{x})}{1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x})} && (3.25)\\
&= \sum_{n=1}^{N_1} (0 - \ln(1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n))) + \sum_{n=N_1+1}^{N} (\mathbf{w}^\mathsf{T}\mathbf{x}_n - \ln(1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n))) &&
\end{aligned}$$

This is obviously a nonlinear optimization problem. At the minimum the condition

$$\boldsymbol{f}(\mathbf{w}) = \nabla Q(\mathbf{w}) = \frac{\partial Q}{\partial \mathbf{w}} = \mathbf{0} \tag{3.26}$$

needs to hold. To determine these zeros, we apply Newton's method. If we have approximate values of $\mathbf{w}^{(0)}$ we can iteratively get improved $\mathbf{w}$ from:

$$\begin{aligned}
\mathbf{w}^{(\nu+1)} &= \mathbf{w}^{(\nu)} - \Delta\mathbf{w}^{(\nu)} && (3.27)\\
\Delta\mathbf{w}^{(\nu)} &= \nabla(\boldsymbol{f}(\mathbf{w}^{(\nu)}))^{-1}\boldsymbol{f}(\boldsymbol{w}^{(\nu)}) = \boldsymbol{H}^{-1}\left(Q(\mathbf{w})|_{w^{(\nu)}}\right)\nabla Q(\mathbf{w})|_{w^{(\nu)}} &&
\end{aligned}$$

where $\boldsymbol{H}(Q)$ is the Hessian of the second derivatives of Q to $\mathbf{w}$ and ∇Q are the vector of first derivatives.

First, we get the first derivatives

$$\nabla_w Q(\mathbf{w}) = \sum_{n=1}^{N_1} \mathbf{x}_n \left(0 - \frac{\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n)}{1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n)}\right) + \sum_{n=N_1}^{N} \mathbf{x}_n \left(1 - \frac{\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n)}{1+\exp(\mathbf{w}^\mathsf{T}\mathbf{x}_n)}\right). \tag{3.28}$$

With the probabilities

$$p_{ni}(\mathbf{w}) = P(\omega_i|\mathbf{x}_n, \mathbf{w}) = \begin{cases} \frac{1}{1+\exp\mathbf{w}^\mathsf{T}\mathbf{x}_n} = \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_n), \text{ if } i = 1 \\ \frac{\exp\mathbf{w}^\mathsf{T}\mathbf{x}_n}{1+\exp\mathbf{w}^\mathsf{T}\mathbf{x}_n} = 1 - \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}) = \sigma(-\mathbf{w}^\mathsf{T}\mathbf{x}), \text{ if } i = 2 \end{cases} \tag{3.29}$$

this is

$$\nabla_w Q(\mathbf{w}) = \sum_{n=1}^{N_1} -\mathbf{x}_n p_{n2}(\mathbf{w}) + \sum_{n=N_1+1}^{N} \mathbf{x}_n p_{n1}(\mathbf{w}) \tag{3.30}$$

Due to $\sigma'(x) = \sigma(x)(1-\sigma(x))$ the second derivatives are

$$\begin{aligned}
\boldsymbol{H}(Q(\mathbf{w})) &= \sum_{n=1}^{N_1} -\frac{\partial p_{n2}(\mathbf{w})}{\partial \mathbf{w}}\mathbf{x}_n + \sum_{n=N_1+1}^{N} \frac{\partial p_{n2}(\mathbf{w})}{\partial \mathbf{w}}\mathbf{x}_n && (3.31)\\
&= \sum_{n=1}^{N_1} -\mathbf{x}_n\mathbf{x}_n^\mathsf{T} p_{n1}(1-p_{n1}) - \sum_{n=N_1+1}^{N} \mathbf{x}_n\mathbf{x}_n^\mathsf{T} p_{n2}(1-p_{n2}) &&
\end{aligned}$$

and since $p_{n1}(1-p_{n1}) = p_{n2}(1-p_{n2})$ finally

$$H(Q(\mathbf{w})) \quad = \quad -\sum_{n=1}^{N} p_n(1-p_n)\ \mathbf{x}_n\mathbf{x}_n^{\mathsf{T}} \tag{3.32}$$

Since the products $p_n(1-p_n)$ are positive, the 2×2 matrix is negative definite.

Altogether we get the iteration sequence

$$\mathbf{w}^{(\nu+1)} = \mathbf{w}^{(\nu)} + \left(\sum_{n=1}^{N} p_n(\mathbf{w}^{(\nu)})(1-p_n(\mathbf{w}^{(\nu)}))\ \mathbf{x}_n\mathbf{x}_n \right)^{-1} \left(\sum_{n=1}^{N_1} -\mathbf{x}_n p_{n2}(\mathbf{w}^{(\nu)}) + \sum_{n=N_1+1}^{N} \mathbf{x}_n p_{n1}(\mathbf{w}^{(\nu)}) \right) \tag{3.33}$$

The following Matlab program shows the essential steps. For any approximate values we have:

- The slope of the logistic function should decrease with the distance between the mean values

$$w_1 = \frac{1}{\mu_2 - \mu_1} \qquad \text{with} \qquad \mu_i = \frac{1}{N_i}\sum x_{ni},\ i = 1.2 \tag{3.34}$$

- The inflection point x_0 should be at $(\mu_1 + \mu_2)/2$, so $w_0 + w_1 x_0 = 0$ applies

$$w_0 = -w_1 x_0 = -w_1 \frac{\mu_1 + \mu_2}{2} \tag{3.35}$$

```
% learn logistic regression: two classes, one feature
% X1 = vector with features class 1
% X2 = vector with features class 2
% maxiter = maximum number of iterations
function w=lr_2(x1,x2,maxiter);
N1=size(x1,1);N2=size(x2,1);N=N1+N2; % Number of samples
X1=[ones(N1,1),x1];X2=[ones(N2,1),x2];X=[X1;X2]; % X matrices
x1m=mean(x1); x2m=mean(x2); % class means
w1=1/(x2m-x1m); % approximate values
w0=-w1*(x1m+x2m)/2; % slope w1, mid-point between means
w=[w0,w1]';
delta_w=[1,1]'; % initial value
for iter = 1:maxiter % iteration loop
iter=iter
p=zeros(N,1);
for n=1:N1
p(n)=1/(1+exp(+w'*X(n,:)')); %p_n1
end
for n=N1+1:N
p(n)=1/(1+exp(-w'*X(n,:)')); %p_n2
end
p1=p(1:N1);p2=p(N1+1:N);
grad=(-ones(1,N1)*([1-p1,1-p1].*X1)+... % gradient
ones(1,N2)*([1-p2,1-p2].*X2))'
H=X'*diag(p.*(1-p))*X % Hessian
delta_w=H\grad; % solve for increments
if norm(w) > 100 | ...% check for convergence
abs(det(H))< 10^(-8) |...
norm(delta_w) < 10^(-6)
return
end
w=w+delta_w % update w
end
return
```

3.2.4 General Separation Functions

The presented classification method can be generalized in different directions.

- The number of features can be arbitrary. Then for d features we have to use the homogeneous parameter vector and the homogeneous feature vector

$$\mathbf{w}_i = \begin{bmatrix} w_{0i} \\ \boldsymbol{w}_{1i} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}. \tag{3.36}$$

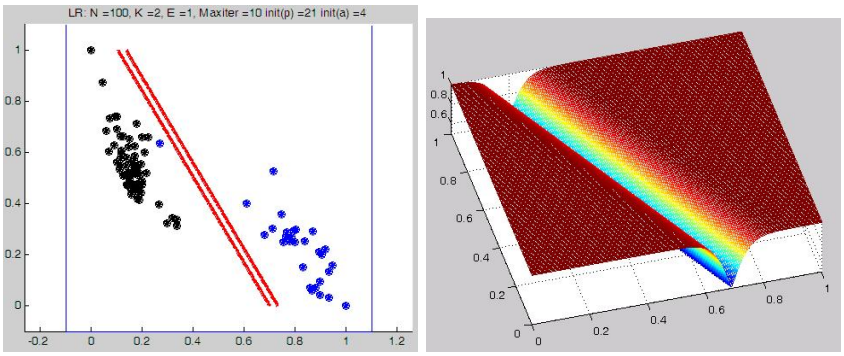


Figure 3.1: Two classes with two features and linear separation. Left: sample and dividing line (double line). Right: maximum a posteriori probability

- The number of classes I can be arbitrary. That does not change the model (3.17). The sum is only to be taken across all classes.

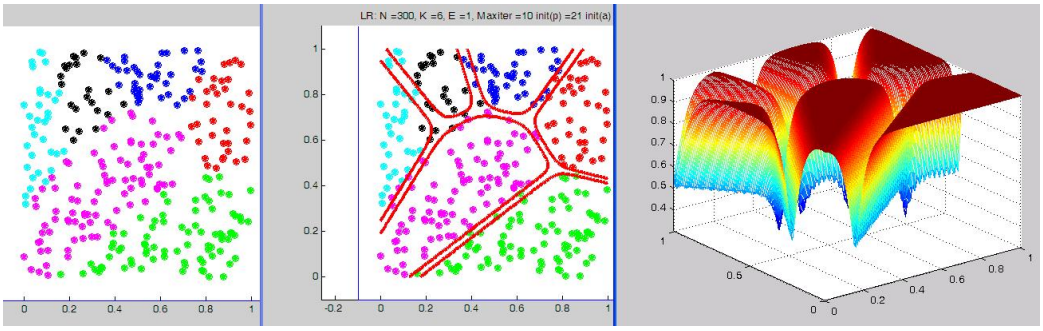


Figure 3.2: Sample (left), separation functions (middle) and maximum a posteriori probability (right) with six classes.

- The strongest limitation is the linearity of the separation functions. This limitation can be removed by transforming the features. For this, we introduce modified features (sometimes called lifted features, due to the extension using polynomials)

$$\phi_k = \phi_k(\boldsymbol{x}) \tag{3.37}$$

For example, if we want to separate three classes with one feature, we choose that transformed features

$$\phi_0 = 1 \qquad \phi_1 = x \qquad \phi_2 = x^2 \tag{3.38}$$

and get the model

$$P(\omega_i \mid x, \mathbf{w}_i) = \frac{\exp(\mathbf{w}_i^{\mathsf{T}} \boldsymbol{\phi}(x))}{\sum_i \exp(\mathbf{w}_i^{\mathsf{T}} \boldsymbol{\phi}(x))} \tag{3.39}$$

with

$$\mathbf{w}_i = \begin{bmatrix} w_0 \\ \boldsymbol{w}_i \end{bmatrix} = \begin{bmatrix} w_{0i} \\ w_{1i} \\ w_{2i} \end{bmatrix} \qquad \boldsymbol{\phi} = \begin{bmatrix} 1 \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} 1 \\ \phi_1 \\ \phi_2 \end{bmatrix} \tag{3.40}$$

so

$$P(\omega_i \mid x, \mathbf{w}_i) = \frac{\exp(w_{0i} + w_{1i}\phi_1(x) + w_{2i}\phi_(x))}{\sum_i \exp(w_{0i} + w_{1i}\phi_1(x) + w_{2i}\phi_(x))} = \frac{\exp(1 + w_{1i}x + w_{2i}x^2)}{\sum_i \exp(1 + w_{1i}x + w_{2i}x^2)} \tag{3.41}$$

If e.g. the elements of class 2 lie within $[-2, +2]$ and the features of the other class to the left and right of it, and therefore one wants to separate at $x = -2$ and at $x = +2$, we can choose the exponent

$$\mathbf{w}^\mathsf{T}\boldsymbol{\phi} = (x-2)(x+2) = -4 + x^2 \tag{3.42}$$

as $\mathbf{w}^\mathsf{T} = [-4, 0, 1]$. The a posteriori probability for class 1 is shown in the figure.

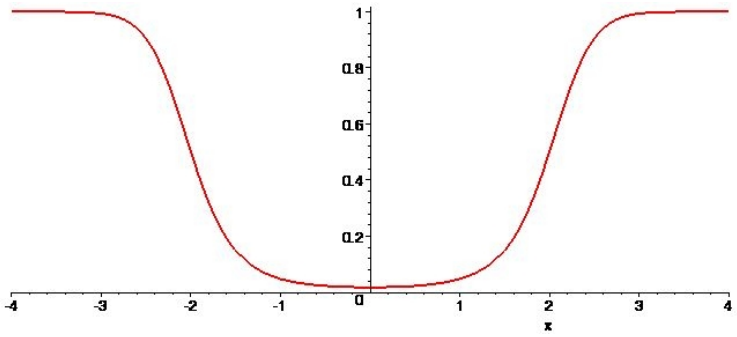


Figure 3.3: A posteriori probability $1/(1 + \exp \mathbf{w}^\mathsf{T}\boldsymbol{\phi})$ for a quadratic separation function $\mathbf{w}^\mathsf{T}\boldsymbol{\phi} = -4 + x^2$.

The figure below shows the learning outcome for four classes, each consisting of a mixed distribution of two Gaussian distributions. The transformed characteristics were ten monomials used up to degree three:

$$\boldsymbol{\phi}^\mathsf{T}(x, y) = [1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3] \tag{3.43}$$

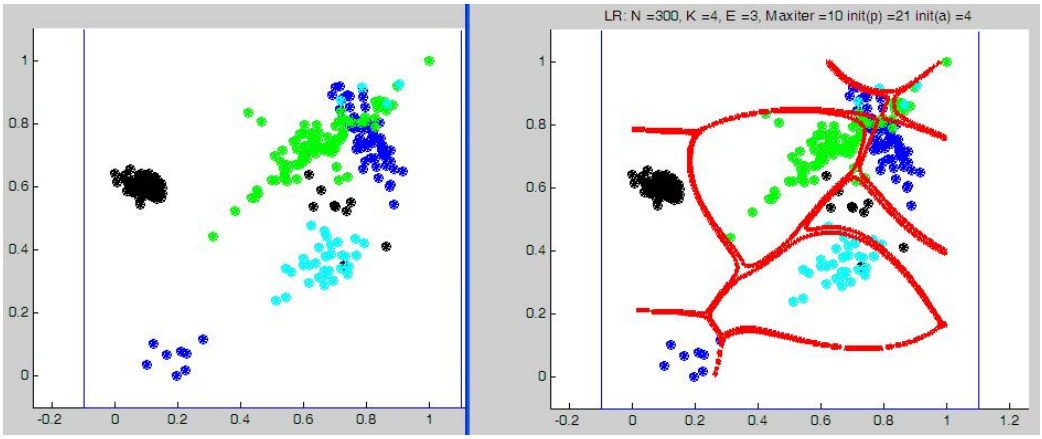


Figure 3.4: Sample (left) and class boundaries (right) for four classes, each representing a mixed distribution of two Gaussian distributions. As transformed features: monomials up to degree three.

3.3 The Strucure of the Hessian for a Symmetric Model

Remark: This part of the note focusses on the structure of the Hessian in case of more than two classes when starting from a symmetric model. Since it is written independently of the previous lecture part, it starts from scratch and partially repeats arguments from the lecture. ◇

We have given a training set $(\boldsymbol{x}, \omega_n)_n, n = 1..., N$ of observed feature vectors $\boldsymbol{x}_n$ of dimension D and their given classes $\omega_n \in \Omega$, with $|\Omega| = K$ being the number of classes. We want to derive a classifier which allows for a new test feature vector $\boldsymbol{x}$ to derive the most probable class ω based on the posterior probability $P(\omega|\boldsymbol{x})$. For this we need to specify the structure of this function $P(\omega|\boldsymbol{x})$ as a function of $\boldsymbol{x}$.

This can be achieved in two ways:

1. Generatively, by using Bayes' rule

$$P(\omega_k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega_k)P(\omega)}{\sum_l P(\boldsymbol{x}|\omega_l)} \tag{3.44}$$

   Then we just need to specify the prior and the likelihood, i. e. the observation model.

2. Discriminatively, by directly modeling the posterior $P(\omega|\boldsymbol{x})$.

We now could rewrite (3.44) in the following way

$$P(\omega_k|\boldsymbol{x}) = \frac{\exp\{\log P(\boldsymbol{x}|\omega_k)P(\omega_k)\}}{\sum_l \exp\{\log P(\boldsymbol{x}|\omega_l)P(\omega_l)\}} \tag{3.45}$$

We gain simplicity, in case $P(\omega|\boldsymbol{x})$ can be written as a linear function of the feature values $\boldsymbol{x}$, e. g. in case all $P(\boldsymbol{x}|\omega_k)$ are Gaussians with the same covariance matrix.

In general, therefore we assume the posterior can be modelled as

$$P(\omega_k|\mathbf{x}, \mathbf{w}_k) = \frac{\exp(\mathbf{w}_k^\mathsf{T}\mathbf{x})}{\sum_l \exp(\mathbf{w}_l^\mathsf{T}\mathbf{x})} \tag{3.46}$$

This type of regression of P on the $D+1$-vector

$$\mathbf{x} = \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix} \tag{3.47}$$

with unknown parameters

$$\mathbf{w}_k = \begin{bmatrix} w_{k0} \\ \boldsymbol{w}_k \end{bmatrix} \tag{3.48}$$

including a constant so-called bias-term w_{k0} is called *logistic regression*.

It results from the special form with $K = 2$ where we can write (3.45)

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1 + \exp\left\{-\log\left[\frac{P(\boldsymbol{x}|\omega_1)}{P(\boldsymbol{x}|\omega_2)}\right] - \log\left[\frac{P(\omega_1)}{P(\omega_2)}\right]\right\}} \tag{3.49}$$

With the logistic function

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \tag{3.50}$$

we observe the a posterior to be a logistic function of the likelihood ratio and the ratio of the priors.

Example: Logistic regression in 1D. Let us assume three classes $K = 3$ separated at $x = -2$ and $x = 1$. This can be achieved by choosing the parameters

$$\mathbf{w}_1 = \begin{bmatrix} -3 \\ -3 \end{bmatrix} \qquad \mathbf{w}_2 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \qquad \mathbf{w}_3 = \begin{bmatrix} 0 \\ 6 \end{bmatrix} \tag{3.51}$$

The three straight lines $y_k(x) = \mathbf{w}_k^\mathsf{T}\mathbf{x}$ or

$$y_1(x) = -3 - 3x \qquad y_2(x) = 5 + x \qquad y_3(x) = 6x \tag{3.52}$$

intersect at $x = -2$ and $x = 1$. The maximum

$$\boxed{y = \max_x(y_k(x)) = \max_x(\mathbf{w}_k^\mathsf{T}\mathbf{x}) = \max_x(-3 - 3x, 5 + x, 6x)\,,} \tag{3.53}$$

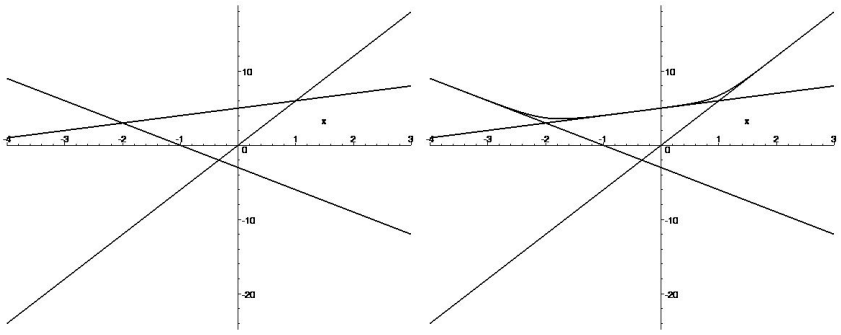


Figure 3.5: Left: maximum of y_k is a convex polygon. Right: replacing the maximum function $\max_x y_k(x)$ by $\log \sum_k \exp y_k(x)$ yields a convex function smoothly approximating the convex polygon.

is a convex polygon, having its maximum for $x \leq 2$ at $y_1(x)$, for $-2 \leq x \leq 1$ at $y_2(x)$ and for $x \geq 1$ at $y_3(x)$.

We write the posterior (3.46) as

$$P(\omega_k|\mathbf{x}, \mathbf{w}_k) = \exp \log P(\omega_k|\mathbf{x}, \mathbf{w}_k) = \exp \left(\underbrace{\mathbf{w}_k^\mathsf{T}\mathbf{x} - \underbrace{\log \sum_l \exp(\mathbf{w}_l^\mathsf{T}\mathbf{x})}_{\log d}}_{\log P} \right) \tag{3.54}$$

and investigate the interior underbraced expression, which is the logarithm $\log d$ of the denominator in (3.46). For this, we compare the polygon with the function

$$\boxed{y = \log \textstyle\sum_k \mathrm{e}^{y_k} = \log \sum_k \mathrm{e}^{\mathbf{w}_k^\mathsf{T}\mathbf{x}} = \log(\mathrm{e}^{-3-3x} + \mathrm{e}^{5+x} + \mathrm{e}^{6x})\,.} \tag{3.55}$$

Obviously we obtain a smoothed version of the polygon.

The logarithm $\log P$ of the posterior in (3.54) is close to 0 in case the straight line $\mathbf{w}_k^\mathsf{T}\mathbf{x} = 0$ is identical to the convex polygon, thus the posterior is close to 1. Otherwise the underbraced expression is negative, thus the posterior is far from 1. In case we would affinely stretch the polygon, e. g. by multiplying the functions y_k with 2, which is equivalent to multiplying the parameters $\mathbf{w}_k$ by that factor, the smooth curve would be closer to the polygon, and the transition of the posterior between the classes would be sharper.

The example gives us an insight into the number of free parameters. A common addend in all exponents does not change the situation, as the polygon is just shifted. Moreover, we may shear the polygon, and set one of the edges to 0, e. g. $y_1(x) = 0$. This can be achieved

$$P(\omega_k|\mathbf{x}, \mathbf{w}_k) = \frac{\exp(-\mathbf{w}_1^\mathsf{T}\mathbf{x})}{\exp(-\mathbf{w}_1^\mathsf{T}\mathbf{x})} \frac{\exp(\mathbf{w}_k^\mathsf{T}\mathbf{x})}{\sum_l \exp(\mathbf{w}_l^\mathsf{T}\mathbf{x})} \tag{3.56}$$

$$= \frac{1}{\sum_l \exp((\mathbf{w}_l - \mathbf{w}_1)^\mathsf{T}\mathbf{x})} \tag{3.57}$$

$$= \frac{1}{1 + \sum_{l=2}^{K} \exp((\mathbf{w}_l - \mathbf{w}_1)^\mathsf{T}\mathbf{x})} \tag{3.58}$$

Thus the problem has only $(D+1)(K-1)$ degrees of freedom.

In the following we want to determine the parameters in the $(D+1) \times K$-matrix

$$W = [\mathbf{w}_1, ..., \mathbf{w}_k, ..., \mathbf{w}_K] \tag{3.59}$$

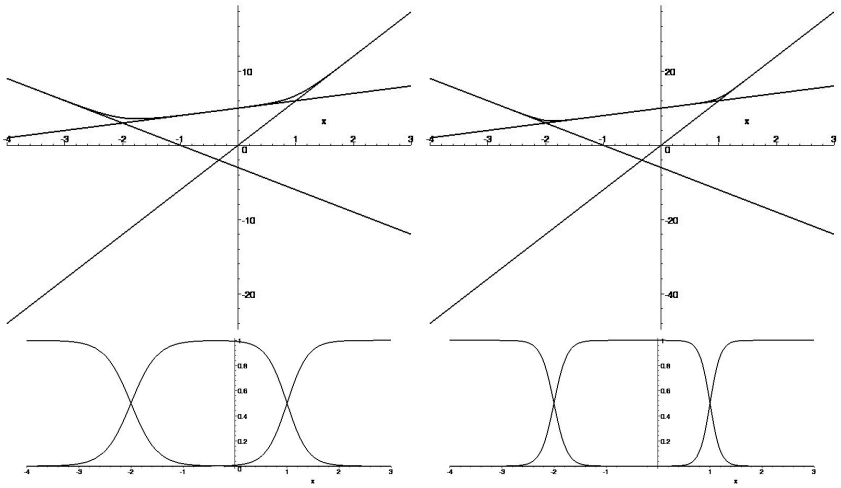


Figure 3.6: Upper left: the logarithm of the denominator of the posterior. Lower left: The posterior for the three classes. Right column: same as fig. 3.6 but with double values $\mathbf{w}_k$. Upper right: the logarithm of the denominator of the posterior. Observe the different scaling in y. Lower right: The posterior for the three classes

from training data. For this we want to maximize the likelihood

$$L(W) = P(\boldsymbol{\omega}, X|W) = \prod_n P(\omega_n, \boldsymbol{x}_n|W) \tag{3.60}$$

with

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \dots \\ \omega_n \\ \dots \\ \omega_N \end{bmatrix} \qquad X = \begin{bmatrix} \boldsymbol{x}_1^\mathsf{T} \\ \dots \\ \boldsymbol{x}_n^\mathsf{T} \\ \dots \\ \boldsymbol{x}_N \end{bmatrix} \tag{3.61}$$

and assuming the samples to be mutually independent. Due to $P(\boldsymbol{\omega}, X|W) = P(\boldsymbol{\omega}|X, W)P(X)$ we also can maximize

$$P(\boldsymbol{\omega}|X, W) = \prod_n P(\omega_n|\boldsymbol{x}_n, W) \tag{3.62}$$

or the logarithm

$$Q(\boldsymbol{W}) = \log P(\omega|X, W) = \sum_n \log P(\omega_n|\boldsymbol{x}_n, W) = \sum_n Q_n(W) \tag{3.63}$$

Assume, in the ν-th iteration we have approximate values $W^{(\nu)}$ for the complete $K \times D$ parameter matrix then an update rule would be

$$W^{(\nu+1)} = W^{(\nu)} - \Delta W^{(\nu)} \tag{3.64}$$

with

$$\Delta W^{(\nu)} = -H^{-1}(Q(W))\, \nabla Q(W) \tag{3.65}$$

where $H(Q(W))$ and $\nabla Q(W)$ are the Hessian and the gradient of Q resp. Eq. (3.65) suggests $H(Q(W))$ to be regular. Actually, it can be shown, that it is negative semidefinite, but singular.

The problem now is: Investigate the structure of the Hessian of the logistic regression, especially its rank and derive an explicit expression for (3.65).

3.3.1 The structure of the gradient and the Hessian

The logarithm of the a posterior is

$$Q(W) = \sum_n Q_n = \sum_n \log P(\omega_n|\boldsymbol{x}_n, W) = \sum_n \left(\boldsymbol{w}_{\omega_n} \boldsymbol{x}_n - \log \left[\sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n \right] \right) \tag{3.66}$$

as only the subvector $\boldsymbol{w}_{\omega_n}$ is relevant for the modelling of the class ω_n.

Thus we obtain the first derivative for each E_n

$$\frac{\partial Q_n(W)}{\partial \boldsymbol{w}_k} = \delta_{k=\omega_n} \boldsymbol{x}_n - \frac{\boldsymbol{x}_n \exp \boldsymbol{w}_k \boldsymbol{x}_n}{\sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n} = \boldsymbol{x}_n \left(\delta_{k=\omega_n} - P(\omega_k|\boldsymbol{x}_n, W) \right) \tag{3.67}$$

being a K-vector. As we are interested in $\frac{\partial Q_n(W)}{\partial \boldsymbol{w}}$ we introduce the $N \times K$-matrix of all conditional probabilities

$$P = \begin{bmatrix} \boldsymbol{p}_1^\mathsf{T} \\ \dots \\ \boldsymbol{p}_n^\mathsf{T} \\ \dots \\ \boldsymbol{p}_N^\mathsf{T} \end{bmatrix} = [p_{nk}] = [P(\omega_k|\boldsymbol{x}_n, W)] \tag{3.68}$$

with the K-vectors $\boldsymbol{p}_n$, containing the current probabilities in the the sample to belong to class k, and can write the partial derivative

$$\frac{\partial Q_n(W)}{\partial \boldsymbol{w}} = (\boldsymbol{e}_{\omega_n} - \boldsymbol{p}_n) \otimes \boldsymbol{x}_n \tag{3.69}$$

being a $K \times D$-vector. The K-vector $\boldsymbol{e}_{\omega_n}$ is the ω_n-th unit vector. The complete gradient therefore is the $K \times D$-vector

$$\nabla_w Q(W) = \frac{\partial Q(W)}{\partial \boldsymbol{w}} = \sum_n \frac{\partial Q_n(W)}{\partial \boldsymbol{w}} = \sum_n (\boldsymbol{e}_{\omega_n} - \boldsymbol{p}_n) \otimes \boldsymbol{x}_n \tag{3.70}$$

The Hessian results from a further differentiation w. r. t. $\boldsymbol{w}$. We find from (3.67)

$$\frac{\partial Q^2(W)}{\partial \boldsymbol{w}_k \partial \boldsymbol{w}_{k'}^\mathsf{T}} = -\frac{\partial \boldsymbol{x}_n P(\omega_k|\boldsymbol{x}_n, W)}{\partial \boldsymbol{w}_{k'}^\mathsf{T}} \tag{3.71}$$

$$= -\boldsymbol{x}_n \frac{\partial P(\omega_k|\boldsymbol{x}_n, W)}{\partial \boldsymbol{w}_{k'}} \tag{3.72}$$

The right factor can be obtained from

$$P(\omega_k|\boldsymbol{x}_n, W) \sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n = \exp \boldsymbol{w}_k \boldsymbol{x}_n \tag{3.73}$$

by differentiation of both sides

$$\frac{\partial P(\omega_k|\boldsymbol{x}_n, W)}{\boldsymbol{w}_{k'}^\mathsf{T}} \sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n + P(\omega_k|\boldsymbol{x}_n, W) \frac{\sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n}{\boldsymbol{w}_{k'}^\mathsf{T}} = \frac{\exp \boldsymbol{w}_k \boldsymbol{x}_n}{\boldsymbol{w}_{k'}^\mathsf{T}} \tag{3.74}$$

$$\frac{\partial P(\omega_k|\boldsymbol{x}_n, W)}{\boldsymbol{w}_{k'}} \sum_l \exp \boldsymbol{w}_l \boldsymbol{x}_n + P(\omega_k|\boldsymbol{x}_n, W) \boldsymbol{x}_n^\mathsf{T} \exp(\boldsymbol{w}_{k'} \boldsymbol{x}_n) = \delta_{kk'} \boldsymbol{x}_n^\mathsf{T} \exp \boldsymbol{w}_k \boldsymbol{x}_n \tag{3.75}$$

Thus we have

$$\frac{\partial Q_n^2(W)}{\partial \boldsymbol{w}_k \boldsymbol{w}_{k'}^\mathsf{T}} = -\boldsymbol{x}_n \boldsymbol{x}_n^\mathsf{T} (\delta_{kk'} - P(\omega_{k'}|\boldsymbol{x}_n, W)) P(\omega_k|\boldsymbol{x}_n, W) \tag{3.76}$$

Again we write this in matrix form, leading to

$$\frac{\partial Q_n^2(W)}{\partial \boldsymbol{w}_k \boldsymbol{w}_{k'}^\mathsf{T}} = -(\mathrm{Diag}(\boldsymbol{p}_n) - \boldsymbol{p}_n \boldsymbol{p}_n^\mathsf{T}) \otimes \boldsymbol{x}_n \boldsymbol{x}_n^\mathsf{T} \tag{3.77}$$

Therefore the Hessian is

$$H(Q(W)) = -\sum_n \left(\mathrm{Diag}(\boldsymbol{p}_n) - \boldsymbol{p}_n\boldsymbol{p}_n^\mathsf{T}\right) \otimes \boldsymbol{x}_n\boldsymbol{x}_n^\mathsf{T} \tag{3.78}$$

The final correction can be written as

$$\Delta\boldsymbol{w} = -\left(\sum_n \left(\mathrm{Diag}(\boldsymbol{p}_n) - \boldsymbol{p}_n\boldsymbol{p}_n^\mathsf{T}\right) \otimes \boldsymbol{x}_n\boldsymbol{x}_n^\mathsf{T}\right)^+ \sum_n (\boldsymbol{e}_{\omega_n} - \boldsymbol{p}_n) \otimes \boldsymbol{x}_n \tag{3.79}$$

Now we analyse the rank of the Hessian. First, we observe that

$$(\mathrm{Diag}(\boldsymbol{p}_n) - \boldsymbol{p}_n\boldsymbol{p}_n^\mathsf{T})\mathbf{1}_K = \boldsymbol{p}_n - \boldsymbol{p}_n \cdot 1 = \mathbf{0} \tag{3.80}$$

Therefore, the Matrix

$$B = \mathbf{1}_K \otimes I_D \tag{3.81}$$

is a right null space of the Hessian and the Hessian has maximum rank

$$\mathrm{rk}(H) \leq (K-1)D \tag{3.82}$$

Therefore, we solve for the parameters by

$$\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\kappa} \end{bmatrix} = \begin{bmatrix} H(Q) & B \\ B^\mathsf{T} & 0_{D\times D} \end{bmatrix}^{-1} \begin{bmatrix} \nabla Q \\ \mathbf{0}_{D\times 1} \end{bmatrix} \tag{3.83}$$

Second, in case the feature vectors are normalized such that

$$\boldsymbol{x}_n^\mathsf{T}\mathbf{1}_D = 0 \qquad \text{for all } n \tag{3.84}$$

a right null space is

$$C = I_K \otimes \mathbf{1}_D \tag{3.85}$$

In this case the rank of the Hessian is

$$\mathrm{rk}(H) \leq (K-1)(D-1) \tag{3.86}$$

3.3.2 The case for two classes

We derive the equations for the two-class problem, hence, $K = 2$, starting at (3.59). We, again, assume the N observed features to be sorted, such that the first N_1 belong to class $\omega_n = 1$ and the second N_2 belong to class $\omega_n = 2$, hence:

$$\boldsymbol{\omega} = [\omega_n] = \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{bmatrix}, \quad X = \begin{bmatrix} \boldsymbol{x}_1^\mathsf{T} \\ \boldsymbol{x}_2^\mathsf{T} \end{bmatrix}, \quad \mathbf{w} = \mathrm{vec}(\mathsf{W}) = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}. \tag{3.87}$$

We assume

$$P_{nk}(\mathbf{w}) := P(\omega_k \mid \boldsymbol{x}_n, \mathbf{w}) = \frac{\exp(\boldsymbol{w}_k^\mathsf{T}\mathbf{x}_n)}{\sum_{i=1}^2 \exp(\boldsymbol{w}_i^\mathsf{T}\mathbf{x}_n)} \qquad k = 1, 2\,, \quad n = 1, \ldots, N\,. \tag{3.88}$$

with the 2-vectors

$$\boldsymbol{p}_n = \tag{3.89}$$

Then we have the complete model:

$$P(\boldsymbol{\omega} \mid \boldsymbol{x}, W) = P(\boldsymbol{\omega}_1 \mid \boldsymbol{x}_1, W)\, P(\boldsymbol{\omega}_2 \mid \boldsymbol{x}_2, W)\,. \tag{3.90}$$

The likelihood we want to maximize, is

$$L(\mathbf{w}) = \prod_{n=1}^N P(\boldsymbol{\omega}_n, \boldsymbol{x}_n \mid \mathbf{w})\,. \tag{3.91}$$

Hence the log-likelihood is

$$
\begin{aligned}
Q(\mathbf{w}) &= \sum_{n=1}^{N_1} \log P(\omega_1, \boldsymbol{x}_n \mid \mathbf{w}) + \sum_{n=N_1+1}^{N} \log P(\omega_2, \boldsymbol{x}_n \mid \mathbf{w}) && (3.92)\\
&= \sum_{n=1}^{N_1} \log \frac{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n)}{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)} + \sum_{n=N_1+1}^{N} \log \frac{\exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)}{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)} && (3.93)\\
&= \sum_{n=1}^{N_1} \mathbf{w}_1\mathbf{x}_n + \sum_{n=N_1+1}^{N} \mathbf{w}_2\mathbf{x}_n - \sum_{n=1}^{N} \log(\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)) && (3.94)
\end{aligned}
$$

The necessary condition for the solution is, using

$$
\nabla_w Q(\mathbf{w}) = \frac{\partial Q(\mathbf{w})}{\partial \mathbf{w}^\mathsf{T}} = \mathbf{0}\,. \tag{3.95}
$$

We use

$$
\frac{\partial \log(\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n))}{\partial \mathbf{w}_1^\mathsf{T}} = \frac{\mathbf{x}_n \exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n)}{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)} \quad n = 1, ..., N_1 \tag{3.96}
$$

First, we have the derivatives w.r.t. $\mathbf{w}_1$:

$$
\begin{aligned}
\nabla_{w_1} Q(\mathbf{w}) &= \sum_{n=1}^{N_1} \left(\mathbf{x}_n - \frac{\mathbf{x}_n \exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n)}{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)} \right) && (3.97)\\
&= \sum_{n=1}^{N_1} \mathbf{x}_n \left(1 - \frac{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n)}{\exp(\mathbf{w}_1^\mathsf{T}\mathbf{x}_n) + \exp(\mathbf{w}_2^\mathsf{T}\mathbf{x}_n)} \right) && (3.98)\\
&= \sum_{n=1}^{N_1} (1 - P_{n1}(\mathbf{w}))\mathbf{x}_n && (3.99)
\end{aligned}
$$

Similarly, we have

$$
\nabla_{w_2} Q(\mathbf{w}^\mathsf{T}) = \sum_{n=N_1+1}^{N} (1 - P_{n2}(\mathbf{w}))\mathbf{x}_n\,. \tag{3.100}
$$

4 Forward and Backpropagation in a Neural Network

Multi-layer Neural Networks and Convolutional neural networks play an important role in interpreting signals. We derive the basic relations for fully connected neural networks, especially the Jacobians for the backward propagation used for learning. Examples based on a simple Matlab-implementation demonstrate the usefulness for elementary classification tasks. The corresponding relations for convolutionally connected layers, which usually are defined by valid correlations, show a remarkably simple relation to those of fully connected layers.

4.1 Preface

The first part of the note (2017/2023) results from discussions in a reading group summarizes the forward and backward propagation in a multi-layer neural network, realized and tested in MATLAB. The Jacobians required for convolutional networks are derived and show a remarkable similarity.

4.2 Problem

Given a multi-layer neural network, derive the forward and backward propagation and the Jacobians for the learning of the parameters. The derivation for fully connected layers largely follow the notation in the book (Nielsen, 2017), however is fully described in matrix notation. The derivations are extended for convolutional neural networks. Referring to single intensity images, the relations are highly similar to those for fully connected layers.

4.3 Setup

A neural network is used to propagate information from the neurons n_j^1 of an input layer, the first layer, to the neuron n^L an output layer, the L-th layer, see Fig. 4.1. Each layer consists of a number J_ℓ of neurons n_j^ℓ which get information only from the neurons $n_k^{\ell-1}$ and only send information to the neurons of the next layer. Each neuron n_j^ℓ takes the outputs $a_k^{\ell-1}$ of all neurons of the previous layers, performs a linear transformation, depending on weights and an additive term, called bias, and applies an activation function f.

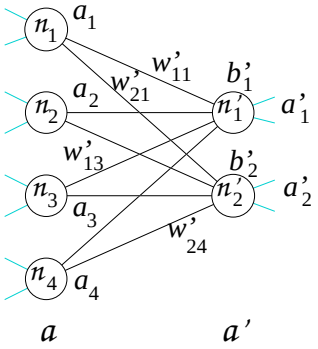


Figure 4.1: Neural network. The figure can be interpreted as a neural network with two layers or a subsection of two consecutive layers in a larger network; in this case we would have given the neurons a superscript indicating the number of the layer they belong to, e.g., $\ell-1$ and ℓ. The first layer has four neurons $n_k^1 =: n_k$. It sends values $\boldsymbol{a} = [a_1, a_2, a_3, a_4]^\mathsf{T}$ to the next layer with two neuron $n_j^1 =: n'_j$. Each output value is *weighted* by w'_{jk} depending where it is sent from (k) and where it is sent to (j). The weights refer to the edges in the graph, some weights are shown in the figure. The prime at the weight w'_{jk} indicates, that it is the decision of the active neuron n'_j how to weight its input. The additive constant, called the *bias* b'_j, is only related to the processing neuron a'_j, thus related to the nodes in the graph. It sets the level of the output of that neuron n'_j. An *activation function* decides how much of the *weighted output* $\boldsymbol{z}' = W'\boldsymbol{a} + \boldsymbol{b}'$ or $z'_j = \sum_k w'_{jk}a_k + b'_j$ is taken as output of the network or sent to the next layer (indicated by thin lines). Thus the transfer between the two layers is $a'_j = f(\sum_k w'_{jk}a_k + b'_j)$. If the function f is taken element wise we also have $\boldsymbol{a}' = f(\boldsymbol{z}') = f(W'\boldsymbol{a} + \boldsymbol{b}')$. The weights W' and the biases $\boldsymbol{b}'$ are unknown and are to be learned from training data. Observe, there are weights and biases related to all layers except the input layer

The activation a_j^ℓ of the j-th neuron in the ℓ-th layer is related to the activations in

the $(\ell-1)$-th layer by (see (Nielsen, 2017, Eq. (23)))

$$a_j^\ell = f\left(\sum_k w_{jk}^\ell a_k^{\ell-1} + b_j^\ell\right) . \tag{4.1}$$

Using vectors and matrices

$$\underset{J_\ell\times 1}{\boldsymbol{a}^\ell} = [a_j^\ell]_{j=1,\ldots,J_\ell} \qquad \underset{J_\ell\times 1}{\boldsymbol{b}^\ell} = [b_j^\ell]_{j=1,\ldots,J_\ell} \qquad \underset{J_\ell\times J_{\ell-1}}{W^\ell} = [w_{jk}^\ell]_{j=1,\ldots,J_\ell;k=1,\ldots,J_{\ell-1}} . \tag{4.2}$$

we also have (see (Nielsen, 2017, Eq. (25)))

$$\boldsymbol{a}^\ell = \boldsymbol{a}^\ell(W^\ell, \boldsymbol{b}^\ell; \boldsymbol{a}^{\ell-1}) = f\left(W^\ell \boldsymbol{a}^{\ell-1} + \boldsymbol{b}^\ell\right) . \tag{4.3}$$

We have L layers $1, ..., \ell, ..., L$, starting with $\ell = 1$. Hence, the vector $\boldsymbol{a}^1$ is the input layer, and the vector $\boldsymbol{a}^L$ is the result of the last layer, called the output layer. The result of the output layer is compared to the desired input $\boldsymbol{y}$, being a vector of J_ℓ elements.

There are different choices for the activation function.

4.4 The Activation Function

The activation function $f(x)$ needs to be nonlinear, since otherwise the concatenation of the matrix operations in (4.1) would boil down to a single linear operation. We present some common activation functions together with their derivative, which we need during gradient descent estimation of the parameters.

At least the following activation functions are in use:

- The sigmoid function $\sigma(x)$,
- the ramp function $\mathrm{ReLU}(x)$,
- the leaky or parametric ramp function $\mathrm{PReLU}(x)$, and
- the softplus function $\mathrm{SoftPlus}(x)$.

We discuss their relations and their derivatives, see Fig. 4.2. When appliying activation

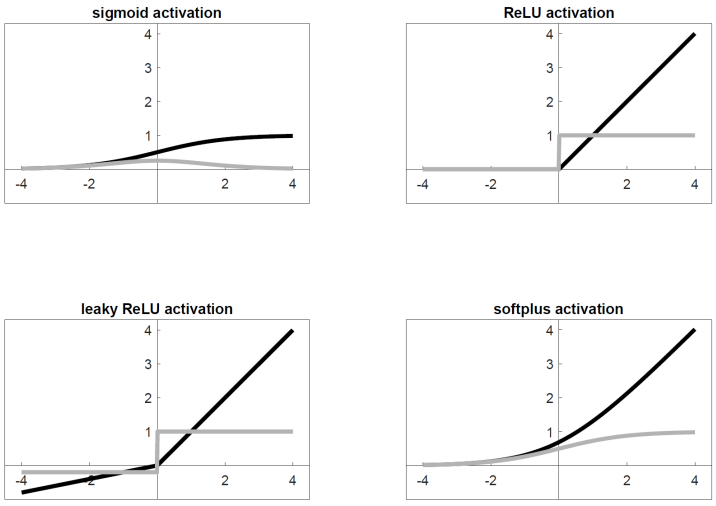


Figure 4.2: Activation functions (black) and their derivatives (gray)

functions to a vector or matrix the function is meant to be used pointwise

$$f(A) = f([a_{ij}]) := [f(a_{ij})] . \tag{4.4}$$

4.4.1 The Sigmoid Function

The oldest activation function is the sigmoid function $\sigma(x)$

$$f(x) := \sigma(x) = \frac{1}{1+\mathrm{e}^{-x}} = \frac{\mathrm{e}^{x}}{1+\mathrm{e}^{x}} = \frac{1}{2}\left(1+\arctan\left(\frac{x}{2}\right)\right) . \tag{4.5}$$

with their derivative

$$\sigma'(x) = \frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} = \sigma(x)(1-\sigma(x)) . \tag{4.6}$$

Since $\exp(x)$ for large x leads to overflow, the sigmoid function does not work for very small negative x. We use the equivalence

$$\sigma(x) = \begin{cases} \dfrac{1}{1+\mathrm{e}^{-x}} \text{ if } x > 0 \\ \dfrac{\mathrm{e}^{x}}{1+\mathrm{e}^{x}} \text{ else} \end{cases} . \tag{4.7}$$

Thus, we redefine

$$\sigma(x) = \frac{\mathrm{e}^{\min(0,x)}}{1+\mathrm{e}^{-|x|}} , \tag{4.8}$$

which works for large positive and negative arguments.

The motivation for using $\sigma(x)$ is, that all arguments are mapped to the interval $[0,1]$, keeping the results numerically stable. The disadvantage is, that the derivative for large values $|x|$ practically is zero, which may hinder the estimation of the parameters within a gradient descent procedure.

4.4.2 The Ramp Function

Often the activation function is chosen to be the *ramp function* $\mathrm{ReLU}(x)$ (rectified linear unit):

$$f(x) := \mathrm{ReLU}(x) = \max(0,x) . \tag{4.9}$$

The ReLU-function has the advantage, that large positive values of x still have an influence onto the learning.

Its first derivative is the step function[1]

$$f'(x) := \mathrm{ReLU}'(x) = s(x) = \frac{1}{2}(1+\mathrm{sign}(x)) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ else} \end{cases} \tag{4.10}$$

Observe, the step function is a crisp version of the sigmoid function:

$$s(x) = \lim_{a\to\infty} \sigma(ax) . \tag{4.11}$$

4.4.3 The leaky or parametrized ramp function

Since the gradient of the ReLU-function is zero for negative values, which causes gradient descent learning to neglect negative arguments, the leaky or parametrized ReLU function PReLU may be used:

$$f(x) := \mathrm{PReLU}(x,p) = \max(px,x) \quad \text{with} \quad 0 < p < 1 . \tag{4.12}$$

It derivative is a vertically shifte step function

$$f'(x) := \mathrm{PReLU}'(x,p) = \frac{\partial \mathrm{PReLU}(x,p)}{\partial x} = (1+p)s(x) - p . \tag{4.13}$$

[1] It is sometimes denoted with $H(x)$. We do not use this naming, since we otherwise come in conflict with the entropy Hx) of a random variable $\underline{x}$.

4.4.4 The SoftPlus Function

Since the ramp function is not differentiable at 0, often the softmax function, a smoothed version of the ramp function, is taken

$$f(x) = \text{SoftPlus}(x) = \ln(1 + \mathrm{e}^x) = -\ln \sigma(-x)\,. \tag{4.14}$$

The last expression can be used for implementation without overflow, see (4.7). Its first derivative is the sigmoid function

$$\text{SoftPlus}'(x) = \sigma(x)\,. \tag{4.15}$$

The difference to the ramp function is small concerning the quality of the result. The ramp function is much faster.

4.5 Forward propagation

We now assume we have more than two layers and want to express the result of the network for a given input, which is called forward propagation.

Given an input $\boldsymbol{x}$ the output $\boldsymbol{y}(\boldsymbol{x})$ of the network is the recursive application of (4.3), assuming $\boldsymbol{a}^1 = \boldsymbol{x}$

$$\boldsymbol{y}(\boldsymbol{x}) := \boldsymbol{a}^L(\boldsymbol{x}) = f\left(W^L\left(f(\ldots W^3 \underbrace{f(W^2\boldsymbol{a}^1 + \boldsymbol{b}^2)}_{\boldsymbol{a}^2} + \boldsymbol{b}^3) + \ldots\right) + \boldsymbol{b}^L\right)\,. \tag{4.16}$$

Observe, the superscripts ℓ of the weights W^ℓ and the biases $\boldsymbol{b}^\ell$ refer to the layer number of the activated layer, thus they start with $\ell = 2$.As iterative procedure we would initiate $\boldsymbol{a}^1 = \boldsymbol{x}$ and then apply (4.3) $L-1$-times, yielding the output $\boldsymbol{y} = \boldsymbol{a}^L$. Hence, we have the iterative algorithm for input $\boldsymbol{x}$:

$$\boldsymbol{a}^1 := \boldsymbol{x}\,, \quad \boldsymbol{a}^\ell = f(W^\ell \boldsymbol{a}^{\ell-1} + \boldsymbol{b}^\ell)\,;\ \text{for } \ell = 2, \ldots, L\,, \quad \text{and} \quad \boldsymbol{y}(\boldsymbol{x}) := \boldsymbol{a}^L\,. \tag{4.17}$$

4.6 Back propagation

The goal of learning is to find weights W^ℓ and biases $\boldsymbol{b}^\ell$ from N training data $(\boldsymbol{x}_n, \boldsymbol{y}_n)_{n=1,\ldots,N}$ (Nielsen (2017) assumes n training data). They should minimize a cost function, e.g., the quadratic costs

$$C(\{W^\ell, \boldsymbol{b}^\ell\}) = \frac{1}{2}\,\frac{1}{N}\sum_{n=1}^{N} ||\boldsymbol{y}_n - \boldsymbol{a}^L(\boldsymbol{x}_n; \{W^\ell, \boldsymbol{b}^\ell\})||^2 = \frac{1}{2}\,\frac{1}{N}\sum_{n=1}^{N} ||\boldsymbol{a}_n^L - \boldsymbol{y}_n||^2\,. \tag{4.18}$$

We made the unknown parameters W^ℓ and $\boldsymbol{b}^\ell$ explicit. We obviously assume the training samples are mutually independent. We may write the cost function as

$$C(\{W^\ell, \boldsymbol{b}^\ell\}) = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N} C_n(\{W^\ell, \boldsymbol{b}^\ell\})\,. \tag{4.19}$$

For this we use a form of steepest descent algorithm. Let all unknown parameters be collected in the vector $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = [\boldsymbol{\theta}^\ell]_{\ell=2,\ldots,L} = \begin{bmatrix} \boldsymbol{w}^\ell \\ \boldsymbol{b}^\ell \end{bmatrix}_{\ell=2,\ldots,L} \quad \text{with} \quad \underbrace{\boldsymbol{w}^\ell}_{W_\ell \times 1} = \text{vec}\, W^\ell\,. \tag{4.20}$$

The number W_ℓ of elements of W^ℓ is

$$W_\ell = J_\ell J_{\ell-1}\,. \tag{4.21}$$

Let us further assume we have an approximate value $\boldsymbol{\theta}^a$ for $\boldsymbol{\theta}$. The steepest decent algorithm iteratively improves $\boldsymbol{\theta}^a$:

$$\boldsymbol{\theta}^{(\nu+1)} = \boldsymbol{\theta}^{(\nu)} - \alpha \left.\frac{\partial C}{\partial \boldsymbol{\theta}^\mathsf{T}}\right|_{\theta=\theta^\nu} . \tag{4.22}$$

Remark: We assume the derivative of a scalar product is $\mathrm{d}(\boldsymbol{a}^\mathsf{T}\boldsymbol{b})/\mathrm{d}\boldsymbol{a}^\mathsf{T} = \boldsymbol{b}$, such that the total differential of the product $\boldsymbol{a}^\mathsf{T}\boldsymbol{b}$ is $\mathrm{d}(\boldsymbol{a}^\mathsf{T}\boldsymbol{b}) = \mathrm{d}\boldsymbol{a}^\mathsf{T}\,\boldsymbol{b} + \boldsymbol{a}^\mathsf{T}\mathrm{d}\boldsymbol{b}$, which is why we differentiate w.r.t. the transposed vector. Hence, $\mathrm{d}(\boldsymbol{a}^\mathsf{T}\boldsymbol{b})/\mathrm{d}\boldsymbol{a} = \mathrm{d}(\boldsymbol{b}^\mathsf{T}\boldsymbol{a})/\mathrm{d}\boldsymbol{a} = \boldsymbol{b}^\mathsf{T}$. ◇

The main task is to determine the partial derivatives

$$\frac{\partial C_n}{\partial \boldsymbol{\theta}^\ell} = \left[\frac{\partial C_n}{\partial \boldsymbol{w}^\ell} \quad \frac{\partial C_n}{\partial \boldsymbol{b}^\ell}\right] , \tag{4.23}$$

which are needed to find the minimum of C w.r.t. w_{jk}^ℓ and b_j^ℓ based on the N observations $\boldsymbol{a}_n^1 = \boldsymbol{x}_n$. We determine them stepwise following the chain rule:

$$\frac{\partial C_n}{\partial \boldsymbol{w}^\ell} = \frac{\partial C}{\partial \boldsymbol{a}_n^L}\frac{\partial \boldsymbol{a}_n^L}{\partial \boldsymbol{a}_n^{L-1}} \cdots \frac{\partial \boldsymbol{a}_n^{\ell+1}}{\partial \boldsymbol{a}_n^\ell}\frac{\partial \boldsymbol{a}_n^\ell}{\partial \boldsymbol{w}^\ell} , \text{ for } \ell = 2, ..., L \tag{4.24}$$

and

$$\frac{\partial C_n}{\partial \boldsymbol{b}^\ell} = \frac{\partial C}{\partial \boldsymbol{a}_n^L}\frac{\partial \boldsymbol{a}_n^L}{\partial \boldsymbol{a}_n^{L-1}} \cdots \frac{\partial \boldsymbol{a}_n^{\ell+1}}{\partial \boldsymbol{a}_n^\ell}\frac{\partial \boldsymbol{a}_n^\ell}{\partial \boldsymbol{b}^\ell} , \text{ for } \ell = 2, ..., L . \tag{4.25}$$

4.6.1 The Derivatives of C w.r.t. $\boldsymbol{a}_n^L$

The cost function is quadratic in $\boldsymbol{a}^L$. Hence we have

$$\underbrace{\frac{\partial C}{\partial \boldsymbol{a}_n^{L\mathsf{T}}}}_{J_\ell \times 1} = \boldsymbol{a}^L(\boldsymbol{x}_n; \{W^\ell, \boldsymbol{b}^\ell\}) - \boldsymbol{y}_n =: \boldsymbol{a}_n^L - \boldsymbol{y}_n . \tag{4.26}$$

4.6.2 The Derivatives of $\boldsymbol{a}^\ell$ w.r.t. W^ℓ, $\boldsymbol{b}^\ell$, and $\boldsymbol{a}^{\ell-1}$

We write the $U_\ell = W_\ell + J_\ell$ parameters

$$\underset{U_\ell \times 1}{\boldsymbol{\theta}^\ell} = \begin{bmatrix} \underset{W_\ell \times 1}{\boldsymbol{w}^\ell} \\ \underset{J_\ell \times 1}{\boldsymbol{b}^\ell} \end{bmatrix} . \tag{4.27}$$

Thus (4.3) reads as (omitting the indices n for the training sample $\boldsymbol{x}_n$)

$$\boldsymbol{a}^\ell = f(\boldsymbol{z}^\ell(\boldsymbol{\theta}^\ell)) , \tag{4.28}$$

with

$$\underset{J_\ell \times 1}{\boldsymbol{z}^\ell(\boldsymbol{\theta}^\ell)} = \underset{J_\ell \times J_{\ell-1} J_{\ell-1} \times 1}{W^\ell} \boldsymbol{a}^{\ell-1} + \underset{J_\ell \times 1}{\boldsymbol{b}^\ell} = \underbrace{\left(\left(\boldsymbol{a}^{\ell-1}\right)^\mathsf{T} \otimes I_{J_\ell}\right)}_{J_\ell \times W_\ell = J_\ell \times (J_{\ell-1} J_\ell)} \underset{W_\ell \times 1}{\boldsymbol{w}^\ell} + \underset{J_\ell \times 1}{\boldsymbol{b}^\ell} . \tag{4.29}$$

Here we used the general rule $\mathrm{vec}(ABC) = (C^\mathsf{T} \otimes A)\mathrm{vec}B$ in the special form $\mathrm{vec}(W\boldsymbol{a}) = \mathrm{vec}(IW\boldsymbol{a}) = (\boldsymbol{a}^\mathsf{T} \otimes I)\mathrm{vec}W$, where the Kronecker product is denoted with $\otimes$.

The Jacobian of $\boldsymbol{a}^\ell$ w.r.t. the parameters $\boldsymbol{\theta}^\ell$, or w.r.t. W^ℓ and $\boldsymbol{b}^\ell$, thus results from the chain rule

$$\underbrace{\frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{\theta}^\ell}}_{J_\ell \times U_\ell} = \underbrace{\frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{z}^\ell}}_{J_\ell \times J_\ell} \underbrace{\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{\theta}^\ell}}_{J_\ell \times U_\ell} = \frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{z}^\ell}\left[\underbrace{\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{w}^\ell}}_{J_\ell \times W_\ell}, \underbrace{\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{b}^\ell}}_{J_\ell \times J_\ell}\right] \tag{4.30}$$

The derivatives $\partial \boldsymbol{a}^\ell / \partial \boldsymbol{z}^\ell$ depend on the choice of the activation function. We generally have

$$\underset{J_\ell \times J_\ell}{\frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{z}^\ell}} = \mathrm{Diag}(f'(\boldsymbol{z}^\ell)\,. \tag{4.31}$$

The derivatives w.r.t. the *weights* are

$$\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{w}^\ell} = (\boldsymbol{a}^{\ell-1,\mathsf{T}} \otimes I_{J_\ell}) \tag{4.32}$$

The derivatives w.r.t. the *biases* are

$$\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{b}^\ell} = I_{J_\ell}\,. \tag{4.33}$$

Finally, we have the Jacobian w.r.t. previous outputs, again using the chain rule

$$\frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{a}^{\ell-1}} = \underbrace{\frac{\partial \boldsymbol{a}^\ell}{\partial \boldsymbol{z}^\ell}}_{J_\ell \times J_\ell} \underbrace{\frac{\partial \boldsymbol{z}^\ell}{\partial \boldsymbol{a}^{\ell-1}}}_{J_\ell \times U_\ell} = \mathrm{Diag}(f'(\boldsymbol{z}^\ell)) W^\ell\,. \tag{4.34}$$

4.6.3 Complete derivatives of C

We now derive expression for the derivatives of C w.r.t. the parameters $\boldsymbol{\theta}^L$, thus need to take the index n into account. They can be written as:

$$\begin{aligned}
\frac{\partial C}{\partial \boldsymbol{\theta}^L} &= \frac{1}{N}\sum_{n=1}^N \frac{\partial C_n}{\partial \boldsymbol{a}_n^L}\frac{\partial \boldsymbol{a}_n^L}{\partial \boldsymbol{z}_n^L}\frac{\partial \boldsymbol{z}_n^L}{\partial \boldsymbol{\theta}^L} \\
&= \frac{1}{N}\sum_{n=1}^N \underbrace{\underbrace{\left(\boldsymbol{a}^L - \boldsymbol{y}_n\right)^\mathsf{T}}_{\partial C_n/\partial \boldsymbol{a}_n^L} \underbrace{\mathrm{Diag}\left(f'(\boldsymbol{z}^L)\right)}_{\partial \boldsymbol{a}_n^L/\partial \boldsymbol{z}_n^L}}_{\boldsymbol{\varepsilon}_n^{L,\mathsf{T}}} \underbrace{\left[(\boldsymbol{a}_n^{L-1,\mathsf{T}} \otimes I_{J_\ell})\,,\, I_{J_\ell}\right]}_{\partial \boldsymbol{z}_n^L/\partial \boldsymbol{\theta}^L}\,.
\end{aligned} \tag{4.35}$$

We first introduce an abbreviation for the first product, (Nielsen (2017) calls it $\boldsymbol{\delta}$, see his equation (30))

$$\boxed{\boldsymbol{\varepsilon}_n^L := \left.\frac{\partial C}{\partial \boldsymbol{z}^{L,\mathsf{T}}}\right|_n = f'(\boldsymbol{z}_n^L) \circ \left(\boldsymbol{a}_n^L - \boldsymbol{y}_n\right)\,.} \tag{4.36}$$

We call it the *L-weighted residual*, since it is weighted with the derivative of the activation function of the L-th input $\boldsymbol{z}^L$.

Hence, using $\boldsymbol{u}^\mathsf{T}(\boldsymbol{v}^\mathsf{T} \otimes I) = (1 \otimes \boldsymbol{u}^\mathsf{T})(\boldsymbol{v}^\mathsf{T} \otimes I) = (\boldsymbol{v} \otimes \boldsymbol{u})^\mathsf{T}$, we obtain

$$\frac{\partial C}{\partial \boldsymbol{\theta}^L} = \frac{1}{N}\sum_{n=1}^N \boldsymbol{\varepsilon}_n^{L,\mathsf{T}} \left[(\boldsymbol{a}_n^{L-1,\mathsf{T}} \otimes I_{J_\ell})\,,\, I_{J_\ell}\right] \tag{4.37}$$

$$= \frac{1}{N}\sum_{n=1}^N \left[(\boldsymbol{a}_n^{L-1} \otimes \boldsymbol{\varepsilon}_n^L)^\mathsf{T} \quad \boldsymbol{\varepsilon}_n^{L,\mathsf{T}}\right]\,. \tag{4.38}$$

This expression can be simplified, if we separate the expressions for the Jacobians for the elements of $W^L = [w_{jk}^L]$ and of $\boldsymbol{b}^L = [b_j^L]$. We have the partial derivative of C w.r.t. w_{jk}^L, which refers to the j-th output ε_{nj}^L at level L and the k-th input a_{nk}^{L-1} at level $L-1$:

$$\frac{\mathrm{d}C}{\mathrm{d}w_{jk}^L} = \frac{1}{N}\sum_{n=1}^N \varepsilon_{nj}^L a_{nk}^{L-1}\,. \tag{4.39}$$

Similarly, we have the partial derivative of C w.r.t. b_j^L

$$\frac{\mathrm{d}C}{\mathrm{d}b_j^L} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_{nj}^L \,. \tag{4.40}$$

In vector notation this is

$$\boxed{\frac{\mathrm{d}C}{\mathrm{d}W^L} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^L \boldsymbol{a}_n^{L-1,\mathsf{T}} \quad \text{and} \quad \frac{\mathrm{d}C}{\mathrm{d}\boldsymbol{b}^L} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^{L\mathsf{T}} \,.} \tag{4.41}$$

Here we have assumed the partial derivative of a scalar to the elements of a matrix has the same size as the matrix.

The Jacobian w.r.t. the parameters $\boldsymbol{a}^{L-1}$ is

$$\frac{\partial C_n}{\partial \boldsymbol{a}_n^{L-1}} = \frac{\partial C_n}{\partial \boldsymbol{z}_n^L}\frac{\partial \boldsymbol{z}_n^L}{\partial \boldsymbol{a}^{L-1}} \tag{4.42}$$

$$= \boldsymbol{\varepsilon}_n^{L,\mathsf{T}} W^L \,. \tag{4.43}$$

Hence, we now can determine the Jacobian w.r.t. $\boldsymbol{\theta}^{L-1}$, which only occur in $\boldsymbol{a}^{L-1}$

$$\frac{\partial C_n}{\partial \boldsymbol{\theta}^{L-1}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial C_n}{\partial \boldsymbol{z}_n^L}\frac{\partial \boldsymbol{z}_n^L}{\partial \boldsymbol{a}_n^{L-1}}\frac{\partial \boldsymbol{a}_n^{L-1}}{\partial \boldsymbol{z}_n^{L-1}}\frac{\partial \boldsymbol{z}_n^{L-1}}{\partial \boldsymbol{\theta}^{L-1}} \tag{4.44}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \underbrace{\frac{\partial C_n}{\partial \boldsymbol{z}_n^L}}_{\boldsymbol{\varepsilon}_n^{L,\mathsf{T}}} \frac{\partial \boldsymbol{z}_n^L}{\partial \boldsymbol{a}_n^{L-1}} \frac{\partial \boldsymbol{a}_n^{L-1}}{\partial \boldsymbol{z}_n^{L-1}} \left[\underbrace{\frac{\partial \boldsymbol{z}_n^{L-1}}{\partial \boldsymbol{w}^{L-1}}}_{J_{\ell-1}\times W_{L-1}} , \underbrace{\frac{\partial \boldsymbol{z}_n^{L-1}}{\partial \boldsymbol{b}^{L-1}}}_{J_{\ell-1}\times J_{\ell-1}} \right] \tag{4.45}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \underbrace{\boldsymbol{\varepsilon}_n^{L,\mathsf{T}} W^L \mathrm{Diag}(f'(\boldsymbol{z}^{L-1}))}_{\boldsymbol{\varepsilon}_n^{L-1,\mathsf{T}}} [(\boldsymbol{a}_n^{L-2,\mathsf{T}} \otimes I_{J_{\ell-1}}) \,,\, I_{J_{\ell-1}}]$$

. (4.46)

We now introduce the $(L-1)$-weighted residual (see (Nielsen, 2017, Eq. (BP2)))

$$\boldsymbol{\varepsilon}_n^{L-1} := \frac{\partial C_n}{\partial \boldsymbol{\theta}^{L-1,\mathsf{T}}} = f'(\boldsymbol{z}_n^{L-1}) \circ \left(W^{L,\mathsf{T}} \boldsymbol{\varepsilon}_n^L \right) \tag{4.47}$$

and obtain

$$\frac{\partial C}{\partial \boldsymbol{\theta}^{L-1}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^{L-1,\mathsf{T}} [(\boldsymbol{a}_n^{L-2,\mathsf{T}} \otimes I_{J_{\ell-1}}) \,,\, I_{J_{\ell-1}}] \tag{4.48}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \left[(\boldsymbol{a}_n^{L-2} \otimes \boldsymbol{\varepsilon}_n^{L-1})^{\mathsf{T}} \;,\; \boldsymbol{\varepsilon}_n^{L-1,\mathsf{T}} \right] . \tag{4.49}$$

Elementwise we have

$$\frac{\mathrm{d}C}{\mathrm{d}w_{jk}^{L-1}} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_{nj}^{L-1} a_{nk}^{L-2} \,. \tag{4.50}$$

and

$$\frac{\mathrm{d}C}{\mathrm{d}b_j^{L-1}} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_{nj}^{L-1} \,. \tag{4.51}$$

Analogously, we have the derivatives w.r.t. $\boldsymbol{\theta}^\ell$

$$\frac{\partial C}{\partial \boldsymbol{\theta}^\ell} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial C}{\partial \boldsymbol{a}_n^L}\frac{\partial \boldsymbol{a}_n^L}{\partial \boldsymbol{a}_n^{L-1}} \cdots \frac{\partial \boldsymbol{a}_n^{\ell+1}}{\partial \boldsymbol{a}_n^\ell}\frac{\partial \boldsymbol{a}_n^\ell}{\partial \boldsymbol{\theta}^\ell} \quad \text{with} \quad \ell =, 2, ..., L \,. \tag{4.52}$$

or

$$\boxed{\frac{\mathrm{d}C}{\mathrm{d}W^\ell} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^\ell \boldsymbol{a}_n^{\ell-1,\mathsf{T}} \quad \text{and} \quad \frac{\mathrm{d}C}{\mathrm{d}\boldsymbol{b}^{\ell\mathsf{T}}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^\ell \,.} \tag{4.53}$$

or individually

$$\frac{\mathrm{d}C}{\mathrm{d}w_{jk}^\ell} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_{nj}^\ell a_{nk}^{\ell-1} \quad \text{and} \quad \frac{\mathrm{d}C}{\mathrm{d}b_j^\ell} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_{nj}^\ell \,. \tag{4.54}$$

with the recursive definition of the residuals

$$\boxed{\boldsymbol{\varepsilon}_n^\ell = f'(\boldsymbol{z}_n^\ell) \circ \left(W^{\ell+1,\mathsf{T}} \boldsymbol{\varepsilon}_n^{\ell+1}\right)} \tag{4.55}$$

initiated by (4.36) at the highest level L.

For getting all derivatives, we need

1. in the forward propagation determine the inputs for each layer.
2. in a backward recursion determine the ℓ-weighted residuals, and finally
3. determine the Jacobians w.r.t. to the weights W^ℓ and to the biases $\boldsymbol{b}^\ell$

4.7 Updating the Parameters

The goal is to update the parameters $\boldsymbol{\theta}$ as a function of the residuals $\boldsymbol{y}_n - \boldsymbol{a}^L$ following (4.22)

$$\boldsymbol{\theta}^{(\nu+1)} = \boldsymbol{\theta}^{(\nu)} - \alpha \left.\frac{\partial C}{\partial \boldsymbol{\theta}^\mathsf{T}}\right|_{\theta=\theta^\nu} \,. \tag{4.56}$$

For the weights and the biases, we thus have

$$W^{\ell,(\nu+1)} = W^{\ell,(\nu)} - \alpha \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^\ell \boldsymbol{a}_n^{\ell-1,\mathsf{T}} \,, \quad \ell = L, ..., 2 \,. \tag{4.57}$$

and

$$\boldsymbol{b}^{\ell,(\nu+1)} = \boldsymbol{b}^{\ell,(\nu)} - \alpha \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\varepsilon}_n^\ell \,, \quad \ell = L, ..., 2 \,. \tag{4.58}$$

On a network with a large number U of parameters, we need a very large number N of training data. The computational complexity for one iteration for homogeneous networks, i.e., networks with the same number J of nodes in each layer, is approximately $O(LJ^2N)$. This is prohibitive. The main factor is N.

Therefore, the steepest decent algorithm is modified in the following manner: In each iteration a random sample of size N' of the training data, a *minibatch*, is taken and the parameters are updated. The update will – hopefully – lead into the correct direction, such that with not to many iterations, the iteration sequence converges. This method is called *stochastic gradient method*.

In order to guarantee that after using N samples in packages of N' all training data are used, it is reasonable to partition the N training data in $K_0 = \lfloor N/N' \rfloor$ minibatches of size N'. Passing through the training samples once is called an *epoch*. The next epochs are built on the basis of a randomly shuffled training data set. The number of epochs is to be specified. If there are E epochs, we use $K = E\lfloor N/N' \rfloor$ minibatches It should be large enough that the procedure converges.

The choice of the learning rate α is tricky. Values between 0.0001 and 1 are reasonable, so starting with $\alpha = 0.01$ is fine.

4.7.1 Algorithm with Loop on Minibatch Elements

We assume we have N data, we require that E epochs with minibatch sizes of N' are applied.

The algorithms thus has the following steps:

1. Initiate the weights and the biases

$$W_{jk}^{\ell} \sim \mathcal{N}(0,1)\,, \quad \ell = 2,\ldots,L \tag{4.59}$$

$$b_j^{\ell} \sim \mathcal{N}(0,1)\,, \quad \ell = 2,\ldots,L \tag{4.60}$$

2. for all E epochs generate $K_0 = \lfloor N/N' \rfloor$ minibatches and update the weights and biases

   (a) Initiate inputs

$$\boldsymbol{a}_n^1 = \boldsymbol{x}_n\,, \quad n = 1,\ldots,N' \tag{4.61}$$

   (b) Perform forward propagation and store the outputs $\boldsymbol{a}_n^{\ell}$ at all levels.

$$\boldsymbol{a}_n^{\ell} = \sigma\left(W^{\ell}\boldsymbol{a}_n^{\ell-1} + \boldsymbol{b}^{\ell}\right)\,, \quad \ell = 2,\ldots,L; n = 1,\ldots,N'\,, \tag{4.62}$$

$$\boldsymbol{y}_n = \boldsymbol{a}_n^L\,, \quad n = 1,\ldots,N'\,. \tag{4.63}$$

   (c) In a backward loop update the weights and the biases and determine the weighted residuals. Start with the weighted residuals

   i. if $\ell = L$

$$\boldsymbol{\varepsilon}_n^L = \sigma'(\boldsymbol{z}_n^L) \circ (\boldsymbol{a}_n^L - \boldsymbol{y}_n) \quad n = 1,\ldots,N' \tag{4.64}$$

   ii. else

$$\boldsymbol{\varepsilon}_n^{\ell} = \boldsymbol{a}_n^{\ell} \circ (\mathbf{1} - \boldsymbol{a}_n^{\ell}) \circ \left(W^{\ell+1,\mathsf{T}}\boldsymbol{\varepsilon}_n^{\ell+1}\right)\,, \quad \ell = L-1,\ldots,2; n = 1,\ldots,N' \tag{4.65}$$

   Then update the weights and the biases

$$W^{\ell} \leftarrow W^{\ell} - \alpha\frac{1}{M}\sum_{n=1}^{M}\boldsymbol{\varepsilon}_n^{\ell}\boldsymbol{a}_n^{\ell-1,\mathsf{T}}\,, \qquad \ell = L,...,2; n = 1,\ldots,N' \tag{4.66}$$

$$\boldsymbol{b}^{\ell} \leftarrow \boldsymbol{b}^{\ell} - \alpha\frac{1}{M}\sum_{n=1}^{M}\boldsymbol{\varepsilon}_n^{\ell\mathsf{T}}\,, \qquad \ell = L,...,2; n = 1,\ldots,N'\,. \tag{4.67}$$

4.7.2 Ridge regression or L_2 regularization

If we want the weights not to get too large, we might add a term in the cost function which penalizes large parameters. For this we rewrite (4.19) we now have

$$C(\{W^{\ell},\boldsymbol{b}^{\ell}\}) = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}C_n(\{W^{\ell},\boldsymbol{b}^{\ell}\}) = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}C_n(\boldsymbol{w},\boldsymbol{b})\,. \tag{4.68}$$

where the vectors $\boldsymbol{w}$ and $\boldsymbol{b}$ contain the weights and biases of all layers. Now we extend this cost function by a quadratic term in the parameters and obtain the regularizing cost function

$$C'(\{W^{\ell},\boldsymbol{b}^{\ell}\}) = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}C_n(\boldsymbol{w},\boldsymbol{b}) + \frac{\lambda}{2}\frac{1}{N}\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\theta}\,. \tag{4.69}$$

Instead of (4.22), we now obtain the update rule

$$\boldsymbol{\theta}^{(\nu+1)} = \boldsymbol{\theta}^{(\nu)} - \alpha\frac{\partial C'}{\partial\boldsymbol{\theta}^{\mathsf{T}}} = \boldsymbol{\theta}^{(\nu)} - \alpha\left(\frac{\partial C}{\partial\boldsymbol{\theta}^{\mathsf{T}}} + \frac{\lambda}{N}\boldsymbol{\theta}^{(\nu)}\right) = \left(1 - \frac{\alpha\lambda}{N}\right)\boldsymbol{\theta}^{(\nu)} - \alpha\left(\frac{\partial C}{\partial\boldsymbol{\theta}^{\mathsf{T}}}\right)\,. \tag{4.70}$$

Experience suggest, that only the weights are to be regularized. Hence, we keep the update of the bias terms (4.67), and replace (4.66) by

$$W^{\ell} \leftarrow \left(1 - \frac{\alpha\lambda}{N}\right)W^{\ell} - \alpha\frac{1}{M}\sum_{n=1}^{M}\boldsymbol{\varepsilon}_n^{\ell}\boldsymbol{a}_n^{\ell-1,\mathsf{T}}\,, \qquad \ell = L,...,2; n = 1,\ldots,N' \tag{4.71}$$

4.7.3 Algorithm with Matrices for Minibatch Elements

For simplifying the notation we collect all N elements belonging to the entities of a mini-batch of training data in a matrix with N columns.

The algorithm can also be written as

1. Initiate the weights and the biases

$$W_{jk}^{\ell} \sim \mathcal{N}(0,1)\,, \quad \ell = 2,\ldots,L \tag{4.72}$$

$$b_{j}^{\ell} \sim \mathcal{N}(0,1)\,, \quad \ell = 2,\ldots,L \tag{4.73}$$

2. for all E epochs generate $K_0 = \lfloor N/N' \rfloor$ mini-batches and update the weights and biases

   (a) Initiate inputs

$$A^1 = X\,, \quad \text{with} \quad X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N'}]\,. \tag{4.74}$$

   (b) Perform forward propagation and store the outputs $\boldsymbol{a}_n^{\ell}$ at all levels.

$$A^{\ell} = f\left(W^{\ell} A_n^{\ell-1} + \boldsymbol{b}^{\ell} \mathbf{1}_N^{\mathsf{T}}\right)\,, \quad \ell = 2,\ldots,L\,. \tag{4.75}$$

$$Y = A^L\,. \tag{4.76}$$

   (c) In a backward loop update the weights and the biases and determine the weighted residuals. Start with the weighted residuals

      i. if $\ell = L$

$$E^L = f'(Z^L) \circ (A^L - Y) \tag{4.77}$$

      ii. else

$$E^{\ell} = f(Z^{\ell}) \circ \left(W^{\ell+1,\mathsf{T}} E^{\ell+1}\right) \overset{f=\sigma}{=} A^{\ell} \circ (1 - A^{\ell}) \circ \left(W^{\ell+1,\mathsf{T}} E^{\ell+1}\right), \quad \ell = L-1,\ldots,2 \tag{4.78}$$

   Then update the weights and the biases

$$W^{\ell} \leftarrow W^{\ell} - \alpha \frac{1}{M} E^{\ell} A^{\ell-1,\mathsf{T}}\,, \qquad \ell = L, ..., 2 \tag{4.79}$$

$$\boldsymbol{b}^{\ell} \leftarrow \boldsymbol{b}^{\ell} - \alpha \frac{1}{M} E^{\ell} \mathbf{1}_N\,, \qquad \ell = L, ..., 2\,. \tag{4.80}$$

4.7.4 Algorithm as Pseudo Code

The pseudo code is given in Algorithm 1. The size of the first and the last layer is given by the dimension of the input and output data. There is no W^2 and $\boldsymbol{b}^2$. [2] We select samples of size $K' = \lfloor N/N' \rfloor$ $\{X_s, Y_s\}$ from the shuffled training data in each epoch $e = 1, ..., E$.

[2] In (LeCun et al., 1998) an n-layer network contains $n-1$ hidden layers and one output layer, thus reflects the fact, that $[W^{\ell}, \boldsymbol{b}^{\ell}]$ only exist for n layers.

Algorithm 1: Back propagation based on training data minimizing $\Omega = ||Y - A^L||^2$;

$[\{W^\ell, \boldsymbol{b}^\ell\}]$=`BackPropagation`$(X, Y, \mathcal{J}, f, f', \alpha, E, N')$

Input: training data $\underset{D_X \times M}{X}$, $\underset{D_Y \times M}{Y}$ with columns of input and output data

Size of hidden layers: $\mathcal{J}_h = (J_2, ..., J_\ell, ..., J_{\ell-1})$

activation function f, its derivative f'

Learning rate: α

Number of epochs: E

Minibatch size: N'

Output: weights $\{\underset{L_\ell \times J_{\ell-1}}{W^\ell}\}$ and biases $\{\underset{J_\ell \times 1}{\boldsymbol{b}^\ell}\}$.

Layers: $\mathcal{J} = (D_X, \mathcal{J}_h, D_Y)$, $L = |\mathcal{J}|$;
Initialize weights: $W^\ell_{jk} \sim \mathcal{N}(0,1)$, $\ell = 2, ..., L, j = 1, ..., J_\ell, k = 1, ..., J_{\ell-1}$;
Initialize biases: $b^\ell_j \sim \mathcal{N}(0,1)$, $\ell = 2, ..., L, j = 1, ..., J_\ell$;
Number of minibatches per epoch: $K' = \lfloor N/N' \rfloor$;
for $e = 1, ..., E$ **do**
    Shuffle training data: $\{X^r, Y^r\} :=$ `random_shuffle`$(\{X, Y\})$;
    for $k = 1, ..., K'$ **do**
        Take N'-sample $\{X_s, Y_s\}$ from $\{X^r, Y^r\}$;
        Forward propagation;
        Initialize: $A^1 := X_s$;
        for $\ell = 2, ..., L$ **do**
            Weighted inputs: $Z^\ell = W^\ell A^{\ell-1} + \boldsymbol{b}^\ell \mathbf{1}_N^\mathsf{T}$;
            Outputs: $A^\ell := f\left(Z^\ell\right)$
        end
        Backward propagation;
        for $\ell = L, ..., 2$ **do**
            if $\ell = L$ **then**
                Residual: $E^L = f'(Z^L) \circ (A^L - Y_s)$
            else
                Residual: $E^\ell := f'(Z^\ell) \circ \left(W^{\ell+1,\mathsf{T}} E^{\ell+1}\right)$
            end
            Update weights: $W^\ell := W^\ell - \alpha/N \; E^\ell A^{\ell-1,\mathsf{T}}$;
            Update biases: $\boldsymbol{b}^\ell := \boldsymbol{b}^\ell - \alpha/N \; E^\ell \mathbf{1}_N$
        end
    end
end

4.8 Examples

4.8.1 Classification with 2D Input Features

The following example shows the possibilities of such a network. In order to easily visualize the results, we assume the objects are described with two features, i.e., points in the plane. We simulate training and test data in three ways, leading to three tapes of generated data:

1. Partitioning the unit square by a Voronoi diagram, with linear region boundaries. The class regions are mutually non-overlapping and convex.
2. Partitioning the unit circle, by transforming the Voronoi diagram using a polar transformation. The class regions are mutually non-overlapping and non-convex.
3. For each class we generate a pair of Gaussian distributions, with random mean vector, random covariance matrix, and random probability. The class regions are overlapping and non-convex.

We start with an example from data type 2.

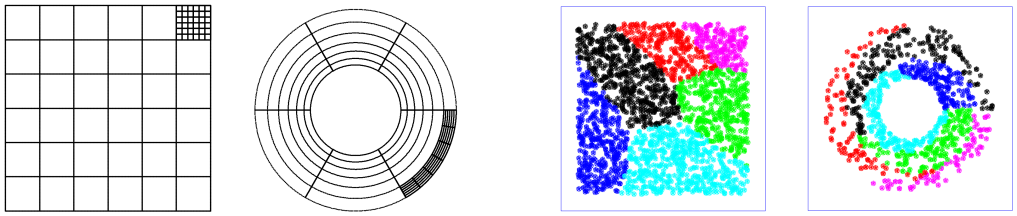
4.8.1.1 Classification with Non-Overlapping 2D Input Features

The data are generated by

- Sampling $C = 6$ points in the unit square representing the centre of each of the 6 classes.
- Sampling $M_{\text{tr}} = 300$ and $M_{\text{te}} = 300$ training and test points $\boldsymbol{x}'_m = [x_{m1}, x_{m2}]^{\mathsf{T}}, m = 1, ..., 600$ in the unit square and performing a nearest neighbourhood classification, see Fig. 4.3c.
- Transforming the data with a (non-linear) polar-transformation (complex exponentiation, see Fig. 4.3)

$$\boldsymbol{x}_m = \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix} = \frac{1}{2\mathrm{e}} \begin{bmatrix} \mathrm{e}^{x'_{m1}} \sin(2\pi x'_{m2}) + \mathrm{e} \\ \mathrm{e}^{x'_{m1}} \cos(2\pi x'_{m2} + \mathrm{e} \end{bmatrix} . \tag{4.81}$$

  The resultant coordinates again are within the unit square.



a.) b.) c.) d.)

Figure 4.3: Non-linear Transformation, From left to right: raw grid, transformed grid, raw data, transformed data (training data)

The neural network has four layers:

- The input layer with 2 neurons, for the x- and y-coordinates
- Two hidden layers with 18 and 12 neurons.
- The output layer with 6 neurons.

We use a minibatch size of $N = 40$ and perform $E = 10\,000$ epochs, which corresponds to $K = EM_{\text{tr}}/N = 75\,000$ iterations. The learning rate is set to $\alpha = 0.1$. The activation function is the sigmoid function. The seed for the random numbers is 15 in the MATLABprogram `test_NN_2_x_c.m`.

The results are shown in Fig. 4.4. The convergence of the optimization is slow, see the upper right subfigure.

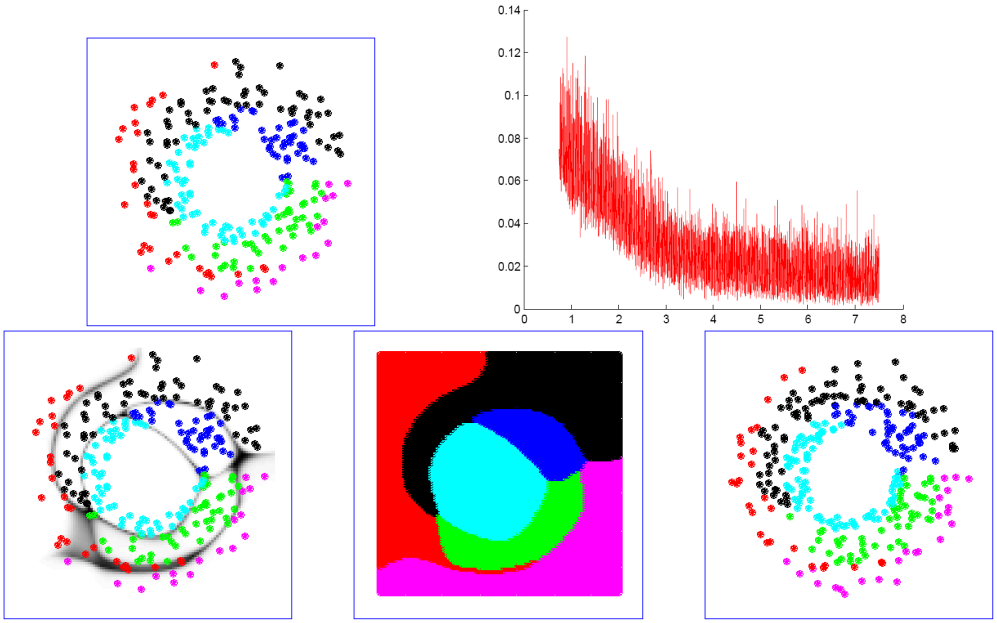


Figure 4.4: Example for learning using a neural network with non-overlapping classes and non-linear class boundaries. Features are the coordinates of the points. We assume 6 classes. The structure of the network is (2-18-12-6), i.e., with two hidden layers with 18 and 12 neurons. **Upper left:** Training data for 6 classes. **Upper right:** Development of the cost function $C(k), k = 7\,5000, ..., 75\,000$ over $E = 10\,000$ epochs with a minibatch size of 40; iteration units $= 10\,000$. **Lower left:** Posterior with training data. **Lower middle:** Regions learned by the neural network. **Lower right:** Test data $N_{\text{te}} = 300$ reach a classification accuracy of 95.25%

When applying the learned network to 100×100 grid points in the unit square, we obtain the acceptance regions for the 6 classes (see the lower left subfigure). They also cover areas where there are no training data, especially at the border and in the interior. The class boundaries appear quite smooth. Also, the thin protrusion of the red class area in the lower part nicely reflects the training data.

The classification of the test data is shown in the lower right subfigure. The classification rate is 96.8%. The classification error of 3.2% reflects the uncertainty in identifying the correct boundaries. This is mainly due to gaps between the classes.

4.8.1.2 Classification with Overlapping 2D Input Features

The following sample shows the learning of the classes for overlapping 2D features.

The data are generated by simulating a mixture of two Gaussian distributions for each class:

- The mean values $\boldsymbol{\mu}_{ci}; c = 1, ..., 5; i = 1, 2$ are sampled from an isotropic Gaussian distribution

$$\underline{\boldsymbol{\mu}_{ci}} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}\right). \tag{4.82}$$

- The covariance matrices $\boldsymbol{\Sigma}_{ci}; c = 1, ..., 6; i = 1, 2$ are sampled using the representation with the PCA of the weight matrix

$$\boldsymbol{\Sigma} = (\boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^{\mathsf{T}})^{-1} \quad \text{with} \quad \boldsymbol{R} = \boldsymbol{R}(\phi) \quad \text{and} \quad \boldsymbol{D} = \text{Diag}([d_1, d_2]) \tag{4.83}$$

The direction ϕ of a principal direction is taken from a uniform distribution $[0, 2\pi]$. The weights d_1 and d_2 in the principal directions are taken from a Gamma-distribution with mean 0.1 and standard deviation 0.1.

- The number of samples is taken from an exponential distribution, and realized by $\underline{n}_c \propto -\log(0.9\,\underline{r})$ with $\underline{r} \sim U(0,1)$. This way obtain a large enough variation of sample sizes per Gaussian and at the same time avoid to small samples per class, see Table 4.1.

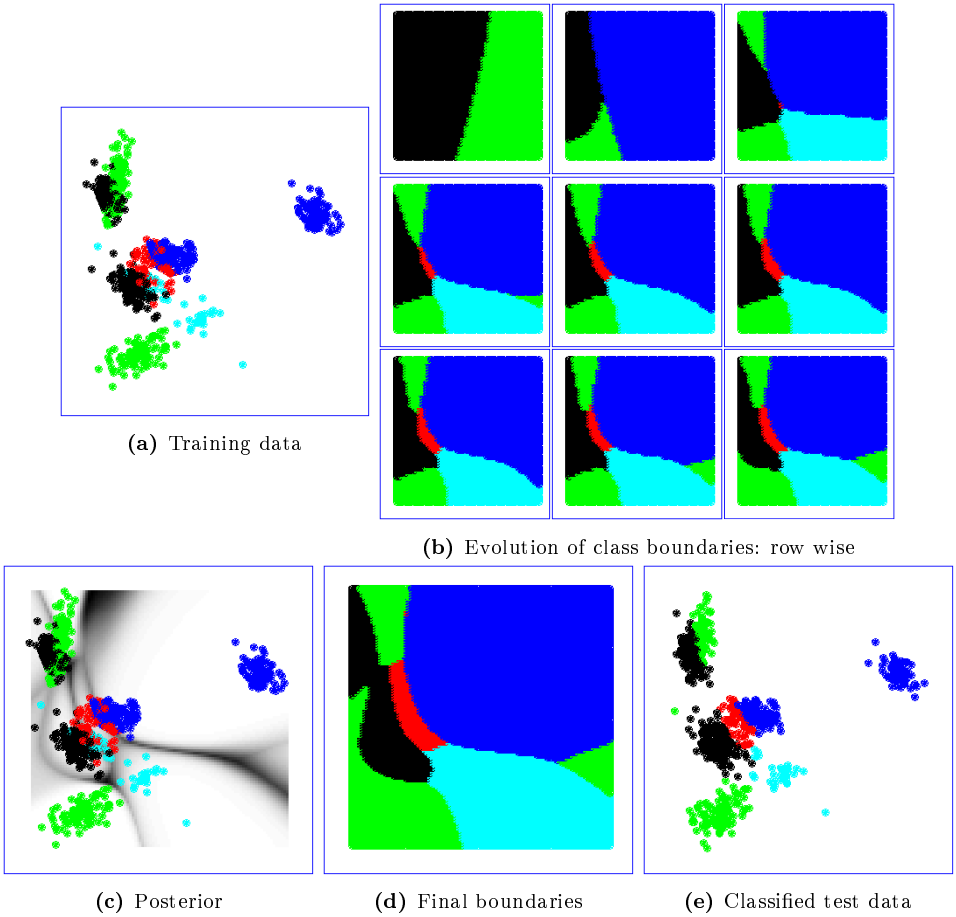
black	blue	green	cyan	red
35.4	23.2	29.3	5.8	6.3

Table 4.1: Probability of the classes

An example is shown in Fig. 4.5

The network has two hidden layers with 25 and 10 neurons, respectively. We run 4000 epochs with a minibatch size of 10. The learning rate is $\alpha = 0.02$.

The results are documented in Figs. 4.5 and Figs. 4.6.

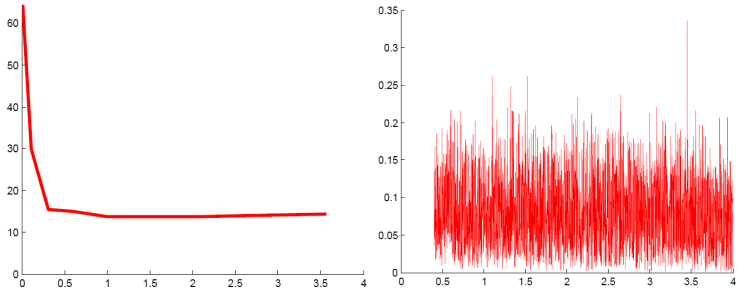


(a) Training data

(b) Evolution of class boundaries: row wise

(c) Posterior **(d)** Final boundaries **(e)** Classified test data

Figure 4.5: Example for learning using a neural network with overlapping class regions. Features are the coordinates of the points. We assume 5 classes. The structure of the network is (2-24-12-5), i.e., with two hidden layers with 18 and 12 neurons. Success rate 84.6%



(a) Failure rate **(b)** Cost function

Figure 4.6: Development of the failure rate and the cost function as a function of the epochs (in hundreds)

- The training data are highly overlapping (see subfigure (a)), e.g., the read class and with the black and the blue class, or the black and the green class. Also observe the single data point of the cyan class on the left side between the two samples of the black class.
- The development of the region boundaries is shown in subfigure (b), to be read from the upper left, row wise to the lower right. Obviously, the network first learns the boundaries between the largest classes. The comparably small red class, overlapping with the large black and blue classes, is represented later. The steps within the iteration scheme and the corresponding failure rates are given in Table 4.2. The

epoch	1.2	11.1	30.9	60.5	100.0	149.4	208.6	277.8	356.8	400
failure rate [%]	64.5	29.9	15.4	15.0	13.8	13.8	13.7	14.1	14.3	15.4

Table 4.2: Failure rates

failure rate drops down to 13.57% after approximately 200 epochs, then slightly increases, see also Fig. 4.6 (a). The development of the cost function does not show a clear decay. Observe, that at the end, the region of the green class breaks out at the left border, obviously trying to catch the few green training data below the upper black cluster.

- The output of the last layer $\boldsymbol{a}^L$ can be interpreted as a posterior $P(y|\boldsymbol{x})$. Applying it to 100×100 grid points in the unit square yields plot in Fig. 4.5 (c). White areas indicate, the the classifier 'thinks' it is good. In the darker areas, especially between the classes the classification is uncertain. As a consequence of the break out of the green class at the left border, the isolated point is classified as green, though it probably should have been classified as cyan.
- The confusion matrix of the six classes, based on 1000 training data, is given in Table 4.3

$P(\hat{y}_c\|y_{c'})$	black	blue	green	cyan	red	$P(y'_c)$
black	94.4	0.0	5.4	0.0	0.3	35.5
blue	0.0	88.4	0.0	0.0	11.6	23.3
green	19.8	0.0	80.2	0.0	0.0	28.8
cyan	28.8	0.0	2.7	57.5	11.0	7.3
red	23.5	11.8	0.0	2.0	62.7	5.1

Table 4.3: Confusion matrix. Conditional probabilities $P(\hat{y}_c|y_{c'})$ given in units of 1%. Number of training data is 1000

4.8.2 Classifying Handwritten Numbers (MNIST)

The images in the MNIST data set have size 28×28. Taking the vector of the images and no, one or two hidden layers and a minibatch size of 10 after 100 epochs yields the results collected in Table 4.4. It also shows the number U of unknown parameters and the running time.

	structure	U	α	failure rate [%]	epochs	CPU time [s]
1	784-10	7 850	0.001	12.7	100	725
2	784-10	7 850	0.0001	11.5	100	572
3	784-30-10	23 860	0.01	10.3	100	900
4	784-100-10	79 510	0.01	7.9	100	1907
5	784-300-10	238 510	0.01	4.5	300	14506
6	784-300-100-10	266 610	0.02	6.0	300	17881

Table 4.4: Results on the MNIST data set. The results of cases 1 and 5 correspond to that was achieved by (LeCun et al., 1998). However, the failure rate for case 6 is worse than 3.05%, that was achieved by (LeCun et al., 1998)

The confusion matrix for the case 1, a linear classifier, is given in Table 4.5. The misclassification probability that the digit 8 is classified as digit 5 is $P(5|8) = 9.1\%$. The next largest misclassification probability is $P(2|3) \approx 8.2\%$.

$P(\hat{y}_c\|y_{c'})$	0	1	2	3	4	5	6	7	8	9	$P(y'_c)$
0	97.6	0.0	0.8	0.2	0.3	0.5	0.4	0.1	0.0	0.1	9.8
1	0.1	98.5	0.5	0.0	0.0	0.2	0.4	0.1	0.1	0.2	11.4
2	0.6	2.4	90.9	0.6	0.8	0.7	1.6	0.6	1.6	0.3	10.3
3	1.0	0.4	8.2	81.5	0.2	5.7	0.4	0.8	0.9	0.9	10.1
4	0.1	0.7	1.6	1.2	90.7	0.1	0.5	0.2	0.3	4.5	9.8
5	2.4	0.3	3.0	5.5	2.6	81.5	2.0	0.3	1.3	1.0	8.9
6	1.5	0.5	4.7	0.3	1.6	2.6	87.9	0.1	0.7	0.1	9.6
7	0.4	1.1	2.6	0.9	2.0	0.7	0.0	86.5	0.3	5.5	10.3
8	1.5	1.8	5.0	5.6	1.6	9.1	1.2	1.1	68.4	4.4	9.7
9	1.1	0.8	0.8	1.2	4.3	1.0	0.0	2.9	0.8	87.2	10.1

Table 4.5: Confusion matrix in percent for case 1, namely the (784-10) network with learning rate $\alpha = 0.001$. The number of test samples is 10 000

The network with the largest number of parameters, network 6 (784-300-100-10), after 300 epochs leads to a misclassification rate of 6%. The maximum misclassification probability $P(4|9 \approx 5\%$.

$P(\hat{y}_c\|y_{c'})$	0	1	2	3	4	5	6	7	8	9	$P(y'_c)$
0	98.7	0.1	0.2	0.3	0.0	0.2	0.4	0.0	0.1	0.0	9.8
1	0.0	98.3	0.5	0.2	0.0	0.1	0.4	0.0	0.4	0.0	11.4
2	1.0	0.3	94.3	1.5	0.5	0.1	0.4	1.0	1.0	0.1	10.3
3	0.0	0.0	1.0	95.6	0.1	1.1	0.1	0.5	1.2	0.4	10.1
4	0.0	0.1	0.6	0.1	95.7	0.0	0.7	0.1	0.8	1.8	9.8
5	0.4	0.1	0.1	1.5	0.3	94.8	1.2	0.3	0.8	0.3	8.9
6	0.4	0.4	0.4	0.1	0.6	0.7	97.0	0.0	0.3	0.0	9.6
7	0.0	1.2	1.7	1.2	0.5	0.0	0.0	93.3	0.2	2.0	10.3
8	0.3	0.2	0.6	1.5	0.6	1.3	0.3	0.5	94.0	0.5	9.7
9	0.4	0.6	0.3	0.9	1.4	0.8	0.1	1.2	1.0	93.4	10.1

Table 4.6: Confusion matrix in percent for case 1, namely the (784-300-10) network with learning rate $\alpha = 0.01$. The number of test samples is 10 000. The failure rate is 4.5%

4.9 Backpropagation with Cross Entropy

Instead of the quadratic loss function we also can use, what is called the cross entropy of the outputs $\boldsymbol{a}_n^L$ of the last layer and the labels $\boldsymbol{y}_n$. For this we assume the $\boldsymbol{y}_n$ are boolean vectors.

The cross entropy of two different distributions of the same variable, with is defined as the expectation of the number of bits to code the variable $\underline{A}$, but using the optimal code for $\underline{B}$:

$$H_A(B) = \mathbb{E}_A(-\log P(B)) = -\sum_i P(\underline{A} = A_i) \log P(\underline{B} = B_i)\,. \tag{4.84}$$

Since the elements of $\boldsymbol{y}_n$ can only take values 0 or 1, here we obtain (applying the logarithm elementwise)

$$C := H_y(a^L) = -\sum_{n=1}^{N} \boldsymbol{y}_n^{\mathsf{T}} \log\left(\boldsymbol{a}_n^L\right) + (1-\boldsymbol{y}_n)^{\mathsf{T}} \log\left(1-\boldsymbol{a}_n^L\right)\,. \tag{4.85}$$

This cost function is minimum if $\boldsymbol{y} = \boldsymbol{a}^L$. Finding the minimum of C can be interpreted as finding the best distribution $(\boldsymbol{a}^L)$ to encode the given labels.

What we need first is the Jacobian of C w.r.t. $\boldsymbol{a}^L$. Taking the ratios element wise, it is given by

$$\begin{aligned}
\frac{\partial C}{\partial \boldsymbol{a}_n^L} &= -\left(\frac{\boldsymbol{y}_n}{\boldsymbol{a}_n^L} - \frac{1-\boldsymbol{y}_n}{1-\boldsymbol{a}_n^L}\right) && (4.86)\\
&= -\left(\frac{\boldsymbol{y}_n \circ (1-\boldsymbol{a}_n^L) - (1-\boldsymbol{y}_n)\circ \boldsymbol{a}_n^L}{\boldsymbol{a}_n^L \circ (1-\boldsymbol{a}_n^L)}\right) && (4.87)\\
&= -\left(\frac{\boldsymbol{y}_n - \boldsymbol{a}_n^L}{\boldsymbol{a}_n^L \circ (1-\boldsymbol{a}_n^L)}\right)\,. && (4.88)
\end{aligned}$$

The L-weighted residual therefore is

$$\begin{aligned}
\boldsymbol{\varepsilon}_n^L &= \frac{\partial C}{\partial \boldsymbol{z}_n^{L\mathsf{T}}} && (4.89)\\
&= -\frac{\sigma'(\boldsymbol{z}_n^L)\circ(\boldsymbol{y}_n - \boldsymbol{a}_n^L)}{\boldsymbol{a}_n^L \circ (1-\boldsymbol{a}_n^L)} && (4.90)\\
&= -\frac{\sigma(\boldsymbol{z}_n^L)(1-\sigma(\boldsymbol{z}_n^L))\circ(\boldsymbol{y}_n - \boldsymbol{a}_n^L)}{\sigma(\boldsymbol{z}_n^L)\circ(1-\sigma(\boldsymbol{z}_n^L))} && (4.91)\\
&= \boldsymbol{a}_n^L - \boldsymbol{y}_n\,. && (4.92)
\end{aligned}$$

Hence, we obtain the Jacobians w.r.t. to W^L and $\boldsymbol{b}^L$

$$\frac{\partial C}{\partial W^{\mathsf{T}}} = \frac{1}{N}\sum_{n=1}^{n} (\boldsymbol{a}_n^L - \boldsymbol{y}_n)\boldsymbol{a}_n^{L-1,\mathsf{T}} \quad \text{and} \quad \frac{\partial C}{\partial \boldsymbol{b}^L} = \frac{1}{N}\sum_{n=1}^{n} (\boldsymbol{a}_n^L - \boldsymbol{y}_n) \tag{4.93}$$

The other partial derivatives remain the same, especially the recursive definition of the ℓ-weighted residual.

4.10 Convolutional connections between layers

The number of parameters $(W^\ell, \boldsymbol{b}^\ell)$ for each layer is ℓ is $J_\ell J_{\ell-1}$, see (4.2). When dealing with images, this number becomes prohibitively large.

We may require, that the weights are

- translation invariant, and
- only refer to a local neighbourhood of a pixel.

These two constraints are fulfilled, if the layer ℓ is formed by a convolution of layer $\ell-1$ with a small kernel. Networks with such a structure are called convolutional neural networks (CNN), or short convolutional networks.

4.10.1 Convolution and Correlation

We repeat parts of the section on Linear Signal Theory. We first address discrete infinite signals, and then specialize to finite signals.

4.10.1.1 General Definition

We refer to the two left subfigures in Fig. 4.7 and first assume the signals have infinite length.

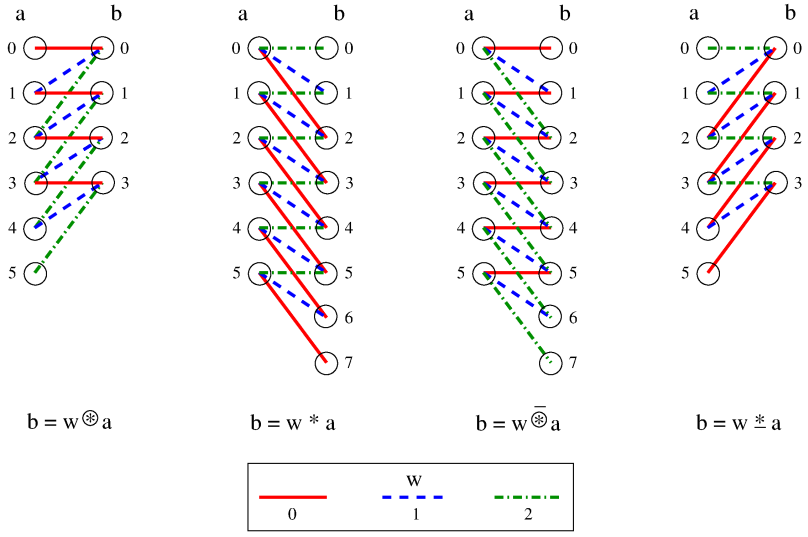


Figure 4.7: Finite convolution and correlation of the 3-kernel $w(i), i = 0, ..., 2$ and the 6-input $a(i), i = 0, ..., 5$ with class `full` and `valid` leading to the output signal $b(i)$: From left to right: **valid correlation** $b = w \circledast a$, **full convolution** $b = w * a$, **valid convolution** $b = w \underline{*} a$, **full correlation** $b = w \overline{\circledast} a$. Not shown: The mode `same` convolution and `same` correlation takes as output a subset of the output of full convolution or of the full correlation with the length identical to the input signal, the selection depending on the implementation

We will refer to the signal $a(i)$ as the input signal and to the signal $w(i)$ as the convolution or correlation kernel, the convolution or correlation filter. The result of the convolution or correlation is the output signal $b(i)$. Most operations are not commutative, why the order of the factors is relevant.

We start with the *correlation*[3] $c = w \circledast a$ of an input signal $a(i)$ with the kernel $w(i)$, both defined on the lattice $\mathbb{Z}$:

$$c = w \circledast a \quad \text{or} \quad c(i) = w(i) \circledast a(i) \tag{4.94}$$

with

$$c(i) = \sum_{k=-\infty}^{+\infty} w(k)a(k+i)\,. \tag{4.95}$$

Correlation describes a moving sum of the values of $a(i+k)$ weighted with the values of $w(k)$. Observe, correlation is not symmetric, thus $w \circledast a \neq a \circledast w$.

A convolution[4] $c(i)$ is defined similarly

$$b = w * a \quad \text{or} \quad b(i) = w(i) * a(i) \tag{4.96}$$

with

$$b(i) = \sum_{k=-\infty}^{+\infty} w(k)a(i-k)\,. \tag{4.97}$$

[3] Sometimes this operation also it is also called cross correlation.

[4] see Sect. ?? on Linear Systems Theory

Observe, convolution is symmetric, thus $w * a = a * w$, which can be seen by setting $k \to i - k'$ and thus $i - k \to k'$ and reversing the summation direction for k'.

Obviously, with the *flipped kernel*

$$w'(i) = w(-i) \tag{4.98}$$

convolution can be described as a moving average weighted with the flipped kernel

$$w(i) * a(i) = w'(i) \circledast a(i) \quad \text{and} \quad w(i) \circledast a(i) = w(-i) * a(i)\,, \tag{4.99}$$

vice cersa, correlation can be described as the convolution with the flipped kernel.

Treating the indices as 2 or higher dimensional vectors leads to two-dimensional, three-dimensional convolution and correlation. All relations then still hold. As an example, with the convolution

$$b(i,j) = w(i,j) * a(i,j) = \sum_{k,l=-\infty}^{+\infty} w(k,l)a(i-k, j-l) \tag{4.100}$$

and the correlation

$$c(i,j) = w(i,j) \circledast a(i,j) = \sum_{k,l=-\infty}^{+\infty} w(k,l)a(k+i, l+j) \tag{4.101}$$

we have

$$c(i,j) = w(i,j) \circledast a(i,j) = w(-i,-j) * a(i,j)\,. \tag{4.102}$$

We also use the function for negating the indices, possibly taking their range into account,

$$w'(i,j) = w(-i,-j) \tag{4.103}$$

to write

$$c(i,j) = w(i,j) \circledast a(i,j) = \text{flip}(w(i,j)) * a(i,j)\,. \tag{4.104}$$

4.10.1.2 Definitions for finite signals

The situation becomes trickier for finite signals $a(i)$ with $i \geq 0$ or $i > 0$. The previous equations cannot be used directly, since negative indices may occur.

We assume the given signals are defined for non-negative indices:

$$a(i),\, i =\in \mathcal{N}_a \quad \text{with} \quad \mathcal{N}_a = [1, N_a - 1] \quad \text{and} \quad |\mathcal{N}_a| = N_a\,, \tag{4.105}$$

and

$$w(i), i \in \mathcal{N}_w \quad \text{with} \quad \mathcal{N}_a = [1, N_a - 1] \quad \text{and} \quad |\mathcal{N}_w| = N_w\,. \tag{4.106}$$

We write convolution in two forms

$$b(i) = w(i) * a(i): \quad b(i) = \sum_k w(k)a(i-k) = \sum_{k+l=i} w(k)a(l)\,. \tag{4.107}$$

Similarly, we write correlation in the two forms

$$b(i) = w(i) \circledast a(i): \quad b(i) = \sum_k w(k)a(i+k) = \sum_{l-k=i} w(k)a(l)\,. \tag{4.108}$$

For finite signals there we can define several modes of convolution, depending on the range of the indices of the output signal $b(i)$ and the relative size N_a of the input signal $a(i)$ and size N_w of the kernel $w(i)$ see Fig. 4.7.

We start with two standard cases, where only sums appear in the definition in order to avoid the access to indices outside the range of the definition of the signals.

1. *Valid correlation.* Correlation in mode `valid` is the default transition for convolutional neural networks. For valid correlation (see Fig. 4.7 right) we use the first form of the definition. The indices appear in all combinations, however, with the constraint that the index $i + k \in \mathcal{N}_a$ must not ly outside the range of the input signal:

$$b(i) = w(i) \circledast a(i) : \quad b(i) = \sum_k w(k)a(i+k) \quad \text{for} \quad N_w \leq N_a \tag{4.109}$$

with

$$i \in \mathcal{N}_b = \{i | i + k \in \mathcal{N}_a, k \in \mathcal{N}_w\} = [0, N_a - N_w] \quad \text{with} \quad |\mathcal{N}_b| = N_w - N_a + 1\,. \tag{4.110}$$

As discussed above, correlation is not commutative. If the size N_w of the kernel is larger than 1, the size of the output is smaller than the size of the input. Moreover, valid correlation is only defined for $N_w \leq N_a$. For $N_w = N_a$ we obtain a single element, namely $b(0)$, which is the scalar product $\boldsymbol{w}^\mathsf{T}\boldsymbol{a}$.

2. *Full convolution.* Convolution in mode `full` occurs as the default operation during backpropagation in convolutional neural networks. For full convolution (see Fig. 4.7 second right) the indices again appear in all combinations without any restrictions on N_a and N_w. Hence, we use the second form of the definition of the convolution:

$$b(i) = w(i) * a(i) : \quad b(i) = \sum_{k+l=i} w(k)a(l) \tag{4.111}$$

with[5]

$$i \in \mathcal{N}_b = \{k + l | k \in \mathcal{N}_w, l \in \mathcal{N}_a\} = [0, N_w + N_a - 2] \quad \text{with} \quad |\mathcal{N}_b| = N_w + N_a - 1\,. \tag{4.112}$$

Since addition and multiplication are commutative, also *full convolution is commutative*, a unique property for finite signals. If the kernel size is larger than 1, the size of the output $b(i)$ is larger than the input $a(i)$.

We now express correlation as convolution and vice versa, and arrive at the following modes of convolution and correlation. For this we use the *flipped kernel*

$$w'(i) = w(N_w - i) \quad \text{with} \quad i \in \mathcal{N}_w\,. \tag{4.113}$$

In contrast to the standard definitions above, we indicate valid convolution and full correlation by an underbar and an overbar.

1. *Full correlation* (see Fig. 4.7 second left) similarly is derived from full convolution

$$b(i) = w(i) \,\overline{\circledast}\, a(i) := w'(i) * a(i) \tag{4.114}$$

or explicitely

$$b(i) = \sum_{k+l=i} w'(k)a(l) = \sum_{k+l=i} w(N_w - k)a(l) \tag{4.115}$$

with

$$i \in \mathcal{N}_b = \{k + l | k \in \mathcal{N}_w, l \in \mathcal{N}_a\} = [0, N_w + N_a - 1]\,. \tag{4.116}$$

2. *Valid convolution* (see Fig. 4.7 left) is derived from valid correlation by exchanging the kernel by the flipped kernel:

$$b(i) = w(i) \,\underline{*}\, a(i) := w'(i) \circledast a(i) \quad \text{for} \quad N_w \leq N_a \tag{4.117}$$

[5] The index set at the same time is the Minkowski sum $\mathcal{N}_w + \mathcal{N}_a$ of the index sets of the kernel and the input

or explicitely

$$\boxed{b(i) = \sum_k w'(k)a(i+k) = \sum_k w(N_w - k)a(i+k)} \tag{4.118}$$

with

$$i \in \mathcal{N}_b = \{i | i + k \in \mathcal{N}_a, k \in \mathcal{N}_w\} = [0, N_a - N_w + 1]\,. \tag{4.119}$$

Valid convolution is not commutative as we see from the constraint of the last summation: when exchanging the order of kernel and input, the constraint would be $k + l = i + N_a$. Also valid convolution is not defined for $N_w > N_a$.

As an example, take

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{a} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 5 \\ 6 \\ 7 \end{bmatrix} \tag{4.120}$$

we obtain

$$\boldsymbol{w} \circledast \boldsymbol{a} = \begin{bmatrix} 5 \\ 12 \\ 13 \\ 19 \end{bmatrix}, \boldsymbol{w} * \boldsymbol{a} = \begin{bmatrix} 3 \\ 2 \\ 7 \\ 9 \\ 8 \\ 17 \\ 12 \\ 14 \end{bmatrix}, \boldsymbol{w} \bar{\circledast} \boldsymbol{a} = \begin{bmatrix} 6 \\ 4 \\ 5 \\ 12 \\ 13 \\ 19 \\ 6 \\ 7 \end{bmatrix}, \boldsymbol{w} \underline{*} \boldsymbol{a} = \begin{bmatrix} 7 \\ 9 \\ 8 \\ 17 \end{bmatrix}. \tag{4.121}$$

Finally, we might define the mode `same` convolution or `same` correlation, where the output signal has the same size as the input signal. This can be based on the result of the full convolution or the full correlation by choosing a subvector having the length of the input signal. This still is not unique: For odd filter kernels we might take the first, last or the middle part of the full output signal.[6]

Symmetric definition of finite `valid` convolution. Obviously, valid convolution results in an output which is a subset of full convolution. This observation may be used to define valid convolution such that it remains commutative. This, then also allows to define correlation in the case when $N_w < N_a$.

We start with full convolution defined as

$$b(i) = w(i) * a(i): \quad b(i) = \sum_{k+l=i} w(k)a(l) \quad \text{for} \quad i \in [0, N_w + N_a - 2] \tag{4.122}$$

The range for the `valid` convolution is $[0, N_a - N_w]$ for $N_w < N_a$. The difference of the two lenghtes is $2(N_w - 1)$. This is why the valid convolution contains the centre part of the full convolution, i.e., omitting $N_w - 1$ elements at both ends. Hence we could have defined valid convolutions as

$$b(i) = \sum_{k+l=i+(N_w-1)} w(k)a(l) \quad \text{for} \quad i \in [0, N_a - N_w] \tag{4.123}$$

[6] Convolution is implemented in MATLAB (funtion `conv(a,b,mode)`) and in Python (function `np.convolve(a,b,mode)`). Both are based on cross correlation by flipping the second argument a. The results partially are different. Python's realization for all modes is commutative, since - after flipping the second argument - internally the two signals are swapped such that the first signal is the longer one implying in mode `same` the output has the length of the longer argument

If we want to define convolution symmetrically, we still would like to take the centre part of the full convolution, which in case $N_a > N_a$ would be

$$b(i) = a(i) \underline{*} w(i): \quad b(i) = \sum_{k+l=i+(N_a-1)} a(k)w(l) \quad \text{for} \quad i \in [0, N_w - N_a] \tag{4.124}$$

We can symmetrize the definition by ensuring then index ranges for both cases are the same, independent of the order of the two factors. Hence the index range should be

$$i \in [0, |N_a - N_w|] \tag{4.125}$$

Then we need to omit

$$D = \frac{1}{2}(N_w + N_a - 2 - |N_a - N_w|) = \min(N_a, N_w) - 1 \tag{4.126}$$

elements on both ends of the full convolution and arrive at the symmetric definition of valid convolution

$$\boxed{b(i) = w(i) \underline{*} a(i): \quad b(i) = \textstyle\sum_{k+l=i+D} w(k)a(l) \quad \text{for} \quad i \in [0, |N_w - N_a|]} \tag{4.127}$$

Since D and the range $\mathcal{N}_b$ are symmetric w.r.t. the sizes N_a and N_w we now have commutativity ensured

$$w(i) \underline{*} a(i) = a(i) \underline{*} w(i)\,. \tag{4.128}$$

Since the above definition of valid correlation only works for $N_w < N_a$, we now may define correlation, such that it works also for $N_a < N_w$, as

$$\boxed{w(i) \circledast a(i): \quad w'(i) \underline{*} a(i) = a(i) \underline{*} w'(i)\,.} \tag{4.129}$$

but still it is not commutative, i.e., $w(i) \circledast a(i) \neq a(i) \circledast w(i)$.

4.10.2 Multi-channel images and filtering

In this context we use matrices where the indices start with 0 in order to simplify equations.

We will distinguish one-channel images, colour images and hyperspectral images. Hence, we have for grey level images

$$\underset{N_i \times N_j}{\mathsf{A}} = [A_{i,j}]\,,\ i = 0, ..., N_i - 1\,, j = 0, ..., N_j - 1 \tag{4.130}$$

for colour images

$$\underset{N_i \times N_j \times 3}{\mathsf{A}} = [\mathsf{A}_c] = [A_{i,j,c}]\,,\ i = 1, ..., N_i - 1\,, j = 0..., N_j - 1\,, c \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}\,, \tag{4.131}$$

and for hyperspectral images[7] [8]

$$\underset{N_i \times N_j \times N_c}{\mathsf{A}} = [A_{i,j,c}]\,,\ i = 0, ..., N_i - 1\,, j = 0, ..., N_j - 1\,, c = 0, ..., N_c - 1\,. \tag{4.132}$$

We will use two-dimensional $M_i \times M_j$ filter kernels $\mathsf{W} = [W(i,j)]$ for two-dimensional images. We will use three dimensional $M_i \times M_j \times 3$ filter kernels $\mathsf{W} = [W(i,j,c)]$ for colour images, and three dimensional $M_i \times M_j \times M_c$ filter kernels $\mathsf{W} = [W(i,j,c)]$ for hyperspectral images, of course allowing $M_c = N_c$.

Though color images and hyperspectral images may be seen as sets of one-channel images, their physical meaning is somewhat different.

Color images, having the three channels red, green and blue, are related to visual perception, even if some colour transformation has been applied. Therefore, it is meaningful to *treat the three channels as a unit*, where the labels or indices are only relevant for perception, thus do not induce an order relation. Hence, even if the indices c (computer internal) are represented by integers, we will not use their order relation.

Hyperspectral images are meant to capture the spectral response as a function of the wavelength. Hence the indices of the channels are related to a physical property, why the *order of the indices c is essential.*

Sometimes we refer to the individual channels, making the interpretation as a list of matrices explicit:

$$\mathsf{A} = [A(i,j,c)] = [\mathsf{A}_c] \quad \text{with} \quad \mathsf{A}_c = [A_c(i,j)] \quad \text{with} \quad i = 0, ..., N_i - 1; j = 0, .., N_j - 1; c = 0, ..., N_c - 1\,. \tag{4.133}$$

Since images always will be finite, in the following we need to take into account three aspects, which are relevant when using convolution operations in neural networks:

1. Correlation as default.
2. Padding of finite images.
3. Skipping rows or columns (*dilation* for increasing the window size.
4. Using a larger step size (*Stride*) to reduce images.
5. Aggregating information (*Pooling*) to reduce images.

In detail we have

[7] Arrays of any number of dimension are called tensors, where the dimension of a tensor often is called the rank of the tensor. This should not to be confused with the rank of a matrix. We often distinguish vectors, matrices and higher (≥ 3) order tensors.

[8] There are two conventions to use brackets. The classical one, used in linear algebra, is to use single brackets for column vectors and matrices, e.g., $\boldsymbol{x} = [x_i]$ and $\mathsf{A} = [A_{ij}]$. Alternatively, as in software packages such as Tensorflow, tensors of rank r are treated as lists of tensors of rank $r - 1$, where rank-1 tensors are row vectors. We stick to the convention of linear algebra.

1. *Correlation:* In the context of convolutional neural networks the definition of the kernels refers to correlation. Instead of the basic definition for infinite images we now have for two-dimensional images the correlation of

$$c(i,j) = w(i,j) \circledast a(i,j)\,, \tag{4.134}$$

with the $M_i \times M_j$-kernel $\mathsf{W} = [w(i,j)]$

$$\begin{aligned} c(i,j) &= \sum_{k=0}^{M_i-1} \sum_{l=0}^{M_i-1} w(k,l)a(k+i,l+j) \\ &\text{for} \quad i \in [0, N_i - M_i]\,,\ j \in [0, N_j - M_j]\,. \end{aligned} \tag{4.135}$$

since $a(i,j)$ is only defined for $0,...,N_i-1$ and $0,...,N_j-1$. When using the flip function (to avoid the interrelation of the prime with the superscript)

$$\mathrm{flip}\,\mathsf{W} = \mathrm{flip}([W_{i,j}]) = [W_{M_i-i,M_j-j}]\,. \tag{4.136}$$

Hence, we have in matrix notation

$$\underset{(N_i-(M_i-1))\times(N_j-(M_j-1))}{\mathsf{C}} = \underset{M_i\times M_j}{\mathsf{W}} \circledast \underset{N_i\times N_j}{\mathsf{A}} = \mathrm{flip}(\mathsf{W}) * \mathsf{A}\,. \tag{4.137}$$

Therefore, if the filter kernel is larger than 1×1, the resulting image C is smaller than the given image A.

2. *Padding:* Since real images have a limited size, the sum's upper and lower limits of the indices need to be taken care of.
We may enlarge the original image, by extending its border, to obtain a filtered image C which has the same size as the input image A. If the filter kernel has an uneven size $2h+1$, we need to extend all four borders by (at least) h. This is called padding. There are several ways to pad, we include the case, where there is no padding, called *valid padding*

   - no padding, called *valid padding*.
   - Extending the image by zero, called *zero-padding*.
   - Extending the image by used specified constant, called *constant-padding*.
   - Extending the image by reflection, i.e., extending the image by a reflected copy of the image, e.g., keeping the left border column and adding the second-left column of the image to the left border. This is also called *reflection padding*.
   - Extending the image by replication, i.e., extending the image by replicating rows and columns, e.g., adding the left column of the image to the left border. This is also called *replication padding*.

   Assume an image is given by the matrix A. Then we obtain the following padded image matrices, see Fig. 4.8

$$\mathsf{A} = \begin{bmatrix} 5 & 4 & 9 & 7 & 5 \\ 6 & 6 & 3 & 8 & 7 \\ 8 & 7 & 4 & 9 & 9 \\ 8 & 7 & 4 & 9 & 9 \end{bmatrix}, \quad \mathsf{A}^{(0)} = \left[\begin{array}{ccc|ccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 5 & 4 & 9 & 7 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 6 & 3 & 8 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 7 & 4 & 9 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 8 & 6 & 4 & 4 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{4.138}$$

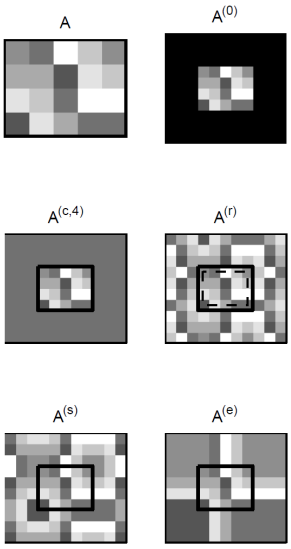


Figure 4.8: Original and padded images. The original image has intensities in the range $[3, 9]$, see (4.138)

$$A^{(\mathrm{c},4)} = \left[\begin{array}{ccc|ccccc|ccc} 4&4&4&4&4&4&4&4&4&4&4\\ 4&4&4&4&4&4&4&4&4&4&4\\ 4&4&4&4&4&4&4&4&4&4&4\\ \hline 4&4&4&5&4&9&7&5&4&4&4\\ 4&4&4&6&6&3&8&7&4&4&4\\ 4&4&4&8&7&4&9&9&4&4&4\\ 4&4&4&3&8&6&4&4&4&4&4\\ \hline 4&4&4&4&4&4&4&4&4&4&4\\ 4&4&4&4&4&4&4&4&4&4&4\\ 4&4&4&4&4&4&4&4&4&4&4 \end{array}\right], \quad A^{(\mathrm{r})} = \left[\begin{array}{ccc|ccccc|ccc} 4&6&8&3&8&6&4&4&4&6&8\\ 9&4&7&8&7&4&9&9&9&4&7\\ 8&3&6&6&6&3&8&7&8&3&6\\ \hline 7&9&4&5&4&9&7&5&7&9&4\\ 8&3&6&6&6&3&8&7&8&3&6\\ 9&4&7&8&7&4&9&9&9&4&7\\ 4&6&8&3&8&6&4&4&4&6&8\\ \hline 9&4&7&8&7&4&9&9&9&4&7\\ 8&3&6&6&6&3&8&7&8&3&6\\ 7&9&4&5&4&9&7&5&7&9&4 \end{array}\right] \tag{4.139}$$

$$A^{(\mathrm{S})} = \left[\begin{array}{ccc|ccccc|ccc} 4&7&8&8&7&4&9&9&9&9&4\\ 3&6&6&6&6&3&8&7&7&8&3\\ 9&4&5&5&4&9&7&5&5&7&9\\ \hline 9&4&5&5&4&9&7&5&5&7&9\\ 3&6&6&6&6&3&8&7&7&8&3\\ 4&7&8&8&7&4&9&9&9&9&4\\ 6&8&3&3&8&6&4&4&4&4&6\\ \hline 6&8&3&3&8&6&4&4&4&4&6\\ 4&7&8&8&7&4&9&9&9&9&4\\ 3&6&6&6&6&3&8&7&7&8&3 \end{array}\right], \quad A^{(\mathrm{e})} = \left[\begin{array}{ccc|ccccc|ccc} 5&5&5&5&4&9&7&5&5&5&5\\ 5&5&5&5&4&9&7&5&5&5&5\\ 5&5&5&5&4&9&7&5&5&5&5\\ \hline 5&5&5&5&4&9&7&5&5&5&5\\ 6&6&6&6&6&3&8&7&7&7&7\\ 8&8&8&8&7&4&9&9&9&9&9\\ 3&3&3&3&8&6&4&4&4&4&4\\ \hline 3&3&3&3&8&6&4&4&4&4&4\\ 3&3&3&3&8&6&4&4&4&4&4\\ 3&3&3&3&8&6&4&4&4&4&4 \end{array}\right] \tag{4.140}$$

3. *Dilation:* In order to increase the kernel size, without increasing the number of parameters the kernel may be dilated, such that the spacing of the addressed input elements is increased. As an example, if the dilation is performed with a factor d we have the dilated correlation

$$\begin{aligned} c(i,j) &= \sum_{k=0}^{M_i-1}\sum_{l=0}^{M_j-1} w(k,l)a(dk+i,dl+j) \\ &\quad \text{for} \quad i=0,...,N_i-dM_i\,,\ j=0,...,N_j-dM_j\,. \end{aligned} \tag{4.141}$$

4. *Stride:* In order to reduce the output image by down sampling, we may proceed as when building image pyramids. The ratio, by which the resulting image is reduced, in the context of CNNs the degree of reduction is called the *stride*. A stride $s = 1$ implies no reduction. A stride $s = 2$ implies a reduction by a factor two. Hence, we might write the correlation with a given stride s as

$$\begin{aligned} c(i,j) &= \sum_{k=0}^{M_i-1} \sum_{l=0}^{M_j-1} w(k,l)a(k+si, l+sj) \\ &\text{with} \quad i = 0, ..., \left\lfloor \frac{N_i - M_i}{s} \right\rfloor \,,\; j = 0, ..., \left\lfloor \frac{N_j - M_j}{s} \right\rfloor \end{aligned} \tag{4.142}$$

   In order to stay flexible, the stride in row and column-direction may be different.

5. *Pooling:* For reducing the size of an image, one possibility is to aggregate the information in a window of the previous layer by a max or mean operation. For a 2×2 window we have the elements

$$A^\ell(i,j) = \max\left(A^{\ell-1}_{2i,2j}, A^{\ell-1}_{2i-1,2j}, A^{\ell-1}_{2i,2j-1}, A^{\ell-1}_{2i-1,2j-1}\right) \,, \tag{4.143}$$

   or the elements

$$A^\ell(i,j) = \frac{1}{4}\left(A^{\ell-1}_{2i,2j} + A^{\ell-1}_{2i-1,2j} + A^{\ell-1}_{2i,2j-1} + A^{\ell-1}_{2i-1,2j-1}\right) \,. \tag{4.144}$$

   of the resulting matrix being reduced by a factor $1/2$.

4.10.3 Multiple filters

It may be meaningful to provide multiple, say M_k, correlation kernels W_k applied to an image, where the filter kernels W_k may be two- or three-dimensional, e.g.,

$$\mathsf{W} = [\mathsf{W}_k] = [W(i,j,c,k] \quad \text{with} \quad \mathsf{W}_k = W_k(i,j,c) \tag{4.145}$$

and

$$i = 0, ..., M_i - 1; j = 0, ..., M_j - 1; c = 0, ..., M_c - 1\,; k = 0, ..., M_k - 1\,. \tag{4.146}$$

For the following discussion we make the following **assumption:** We only handle grey level images, colour images or feature images, no hyperspectral images. Specifically, we assume the third dimension of the filter kernel to have dimension $M_c = 1$ for one channel images and we assume $M_c = N_c$ for color images or $M_c = N_k$ for feature images. Hence the c-dimension of the filtered image will have dimension 1, which we will not make explicit in the result. Therefore, we have the correlation result

$$\mathsf{B} = \mathsf{W} \circledast \mathsf{A} = [B(i,j,k)] \quad \text{with} \quad \mathsf{B}_k = B_k(i,j) = \mathsf{W}_k(i,j,c) \circledast \mathsf{A}(i,j,c)\,. \tag{4.147}$$

Observe, B also is a three-dimensional array, this time the third dimension referring to the feature number k.

Then, padding is applied to the first two indices for each channel individually, and denoted as

$$\overline{\mathsf{A}}_p := \mathrm{padd}(\mathsf{A}, \mathtt{p}) = [\mathrm{padd}(\mathsf{A}_c, \mathtt{p})] \quad \text{or} \quad \mathrm{padd}(\mathsf{A}, p) = [\mathrm{padd}(\mathsf{A}_k, \mathtt{p})]\,. \tag{4.148}$$

where $\mathtt{p}$ denotes the mode of padding.

Hence convolving a three-channel image with a single three-channel kernel ($M_k = 1$), will result in a one-dimensional image.

We are now prepared to describe forward propagation. For the derivations we will restrict to the case

$$M_c = 1 \quad \text{and} \quad N_k = 1\,, \tag{4.149}$$

in order to simplify notation, thus assume one-channel images and a single filter. Then the arrays mentioned up to now are matrices. We follow https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/.

4.10.4 Forward propagation

We will allow for three different ways to come from one layer to the next.

Instead of vectors $\boldsymbol{a}^\ell$ or $\boldsymbol{z}^\ell$ we now have $N^\ell \times N^\ell$ arrays A^ℓ or Z^ℓ.

1. A convolutional connection. We assume no padding. Here we have, when referring to the outputs,

$$\underset{N_i^\ell \times N_j^\ell}{\mathsf{A}^\ell} = f\left(\underset{M_i^\ell \times M_j^\ell}{\mathsf{W}^\ell} \circledast \underset{(N_i^{\ell-1})\times(N_j^{\ell-1})}{\mathsf{A}^{\ell-1}} + \underset{N_i^\ell \times N_j^\ell}{\mathsf{B}^\ell}\right) \tag{4.150}$$

   or, when referring to the inputs,

$$\mathsf{Z}^\ell = \mathsf{W}^\ell \circledast f(\mathsf{Z}^{\ell-1}) + \mathsf{B}^\ell\,. \tag{4.151}$$

   With the number p^ℓ of the additional rows and columns during pooling the matrices of the input layer $\ell - 1$, the sizes of the matrices are related by

$$N_i^\ell = N_i^{\ell-1} - M_i^\ell + 1 \quad \text{and} \quad N_j^\ell = N_j^{\ell-1} - M_j^\ell + 1\,. \tag{4.152}$$

   The correlation kernel is assumed to be of odd size $M_i \times M_j = (2h+1) \times (2h+1)$, usually with $h = 1$, thus a 3×3 kernel.

2. A pooling connection, which allows to reduce the image. We only discuss a special max-pooling with a stride of 2: For non-overlapping 2×2 subimages we take the maximum value. This reads as (see above)

$$A^\ell(i,j) = \max\left(A^{\ell-1}_{2i,2j}, A^{\ell-1}_{2i-1,2j}, A^{\ell-1}_{2i,2j-1}, A^{\ell-1}_{2i-1,2j-1}\right)\,, \tag{4.153}$$

   and leads to a reduction of the image

$$M^\ell = \left\lfloor \frac{M^{\ell-1}}{2} \right\rfloor \quad \text{and} \quad N^\ell = \left\lfloor \frac{N^{\ell-1}}{2} \right\rfloor \tag{4.154}$$

3. A full connection, as above. This usually is applied after an adequate reduction of the image and applied to its vector. Hence, we have

$$\boldsymbol{a}^\ell = f(\mathsf{W}^\ell \boldsymbol{a}^{\ell-1} + \boldsymbol{b}^\ell)\,, \tag{4.155}$$

   with for the transition from convolutional to full connections

$$\underset{(N_i^{\ell-1})(N_j^{\ell-1})}{\boldsymbol{a}^{\ell-1}} = \mathrm{vec}\mathsf{A}^{\ell-1}\,. \tag{4.156}$$

   The size of the layer ℓ can be chosen arbitrarily.

We can specify such a CNN by the sequence of triplets, assuming the kernels are quadratic $3 \times 3 \times_k$:

$$\left[[N_i^1, N_j^1], \{[M_i^\ell, M_j^\ell], t^\ell, f^\ell\}\right] \quad \text{with} \quad t^\ell \in \{\texttt{conv}(M_k^\ell), \texttt{pool}, \texttt{full}\} \quad \text{for} \quad \ell = 2, ..., L \tag{4.157}$$

specifying the size of the layer, the type of connection to this layer, possibly including the number of filter kernels, and the activation function for this layer. Software packages such as Tensorflow or PyTorch allow for more general specifications.

In the following assume the following restrictions:

- We assume only two-dimensional convolutional kernels.
- for a pooling layer, the activation function is $f^\ell(x) = x$, there are no parameters to be estimated.
- for full layers $N^\ell = 1$, and
- there are – up to now – no convolutional layers after a full layer.

We may have several convolutional layers without a pooling layer.

4.10.5 Backpropagation and Jacobians

We minimize a cost function, e.g., the quadratic costs, as above

$$C(\{W^\ell, B^\ell\}) = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N} ||Y_n - A^L(A_n^1; \{W^\ell, B^\ell\})||^2 = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N} ||A_n^L - Y_n||^2 = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N} C_n \,. \tag{4.158}$$

Again, we made the unknown parameters W^ℓ and B^ℓ explicit.

4.10.5.1 Jacobian of convolutional connections

We need the Jacobians of the optimization function w.r.t. the unknown parameters, namely the elements of the matrices W^ℓ, either convolution kernels or transfer matrices, and the element of the bias B, see (4.24) and (4.25), namely

$$\frac{\partial C}{\partial W^\ell} = \left[\frac{\partial C}{\partial W(i,j,c,k)^\ell}\right] \quad \text{and} \quad \frac{\partial C}{\partial B^\ell} = \left[\frac{\partial C}{\partial B(i,j,k)^\ell}\right], \tag{4.159}$$

which have the same size as W^ℓ and B^ℓ.

Jacobian of C w.r.t. W^ℓ. We start with the Jacobian w.r.t. W^ℓ. With the chain rule we have

$$\underset{M_i^\ell \times M_j^\ell}{\frac{\partial C}{\partial W^\ell}} = \left[\frac{\partial C}{\partial W_{mn}^\ell}\right] = \left[\frac{\partial C}{\partial \boldsymbol{z}^\ell}\frac{\partial \boldsymbol{z}^\ell}{\partial W_{mn}^\ell}\right] \tag{4.160}$$

with the vector $\boldsymbol{z} = \mathrm{vec}Z$ of the matrix Z. Observe, the product in the last bracket can be un-vectorized, leading to

$$\left[\frac{\partial C}{\partial W_{m,n}^\ell}\right] = \left[\sum_{i,j}\frac{\partial C}{\partial Z_{i,j}^\ell}\frac{\partial Z_{i,j}^\ell}{\partial W_{m,n}^\ell}\right] = \mathrm{tr}\left(\left[\frac{\partial C}{\partial \left[Z_{i,j}^\ell\right]}\right]^\mathsf{T}\frac{\partial \left[Z_{i,j}^\ell\right]}{\partial W_{m,n}^\ell}\right). \tag{4.161}$$

since generally $\sum_{i,j} U_{i,j}V_{i,j} = (\mathrm{vec}U)^\mathsf{T}\mathrm{vec}V = \mathrm{tr}(U^\mathsf{T}V)$. We use the abbreviations (taken the large greek letter E)

$$\underset{N_i^\ell \times N_j^\ell}{E^\ell} := \frac{\partial C}{\partial Z^\ell} = \left[\frac{\partial C}{\partial Z_{i,j}^\ell}\right] \quad \text{and} \quad \boldsymbol{\varepsilon}^\ell := \mathrm{vec}(E^\ell) = \frac{\partial C}{\partial \boldsymbol{z}^{\ell,\mathsf{T}}}. \tag{4.162}$$

The partial derivative $\partial Z_{i,j}^\ell / \partial W_{m,n}^\ell$ can be derived from

$$[Z_{i,j}^\ell] = \left[\sum_{m',n'} W_{m',n'}^\ell A_{i+m',j+n'}^{\ell-1} + B_{i,j}^\ell\right] \tag{4.163}$$

such that for $m = m'$ and $n = n'$ we have

$$\frac{\partial [Z_{i,j}^\ell]}{\partial W_{m,n}^\ell} = [A_{i+m,j+n}^{\ell-1}]\,. \tag{4.164}$$

since all other summands in (4.163) lead to zero derivative w.r.t. $W_{m,n}$. Now, inserting (4.162) and (4.164) into (4.161) we find

$$\underset{M_i^\ell \times M_j^\ell}{\frac{\partial C}{\partial W^\ell}} = \left[\sum_{i,j}\frac{\partial C}{\partial Z_{i,j}^\ell}\frac{\partial Z_{i,j}^\ell}{\partial W_{m,n}^\ell}\right] = \left[\sum_{i,j} E_{i,j}^\ell A_{i+m,j+n}^{\ell-1}\right] = \underset{N_i^\ell \times N_j^\ell}{E^\ell} \circledast \underset{N_i^{\ell-1} \times N_j^{\ell-1}}{A^{\ell-1}}, \tag{4.165}$$

see (4.53) left, there including all N samples. Observe, we have the size of the correlation matrix, e.g., for the first index

$$N_i^{\ell-1} - (N_i^\ell + 1) = N_i^{\ell-1} - \left(N_i^{\ell-1} - (M_i^\ell + 1) + 1\right) = M_i^\ell, \tag{4.166}$$

see (4.137).

Jacobian of C w.r.t. $\boldsymbol{B}^\ell$. Similarly, we have

$$\frac{\partial C}{\partial \boldsymbol{B}^\ell} = \left[\frac{\partial C}{\partial B_{ij}^\ell}\right] = \left[\frac{\partial C}{\partial \boldsymbol{z}^\ell}\frac{\partial \boldsymbol{z}^\ell}{\partial B_{ij}^\ell}\right] . \tag{4.167}$$

Since all partial derivatives of $\partial Z_{ij}/\partial B_{ij} = 1$ are one (see (4.151)) this expressions simplifies to

$$\frac{\partial C}{\partial \boldsymbol{B}^\ell} = \left[\frac{\partial C}{\partial B_{ij}^\ell}\right] = \left[\frac{\partial C}{\partial Z_{ij}^\ell}\right] = \boldsymbol{E}^\ell . \tag{4.168}$$

see (4.53) right.

Recursive relation. We start from (4.151)

$$\boldsymbol{Z}^\ell \quad = \quad \boldsymbol{W}^\ell \circledast \boldsymbol{A}^{\ell-1} + \boldsymbol{B}^\ell \quad \text{with} \quad \boldsymbol{A}^{\ell-1} = f(\boldsymbol{Z}^{\ell-1}) . \tag{4.169}$$

The partial of the second expression simply is

$$\frac{\partial A_{i,j}^\ell}{\partial Z_{i,j}^\ell} = f'\left(Z_{i,j}^{\ell-1}\right) . \tag{4.170}$$

since all other partial derivatives $\partial A_{i,j}/\partial A_{i',j'}, i \neq i', j \neq j'$ are zero, which we take into account in the following.

Therefore, we are interested in the Jacobian

$$E_{i,j}^{\ell-1} = \frac{\partial C}{\partial Z_{i,j}^{\ell-1}} = \sum_{m,n} \frac{\partial C}{\partial Z_{m,n}^\ell}\frac{\partial Z_{m,n}^\ell}{\partial Z_{i,j}^{\ell-1}} = \sum_{m,n} E_{m,n}^\ell \frac{\partial Z_{m,n}^\ell}{\partial Z_{i,j}^{\ell-1}} \quad \text{for} \quad i \in [0, N_i^{\ell-1}-1], j \in [N_j^{\ell-1)}-1] \tag{4.171}$$

We need to take care of the range of the summation indices: The matrix $\boldsymbol{E}^{\ell-1}$ generally is larger than the matrix $\boldsymbol{E}^\ell$, due to the convolution of $\boldsymbol{E}^{\ell-1}$ with the kernel $\boldsymbol{W}^\ell$. Furthermore, the Jacobian $Z_{m,n}^\ell Z_{i,j}^{\ell-1}$ expresses the effect of changes in $\boldsymbol{Z}^\ell$ due to changes in $\boldsymbol{Z}^{\ell-1}$, which only are local, due to the limited size of the convolution kernel $\boldsymbol{W}^\ell$. Hence, the non-zeros in this Jacobian will be identical to the non-zeros in $\boldsymbol{W}^\ell$. Hence, the range of the summation will only be $[0, M_i^\ell - 1] \times [0, M_j^\ell - 1]$.

The second factor now can be derived from (4.163)

$$Z_{i,j}^\ell = \sum_{m',n'} W_{m',n'}^\ell f(Z_{i+m',j+n'}^{\ell-1}) + B_{i,j}^\ell \tag{4.172}$$

In order to be able to differentiate w.r.t. $Z_{i,j}^{\ell-1}$ we change the index of $Z_{i,j}^\ell$

$$Z_{i-m,j-n}^\ell = \sum_{m',n'} W_{m',n'}^\ell f(Z_{i+m'-m,j+n'-n}^{\ell-1}) + B_{i,j}^\ell \tag{4.173}$$

for all elements of $\boldsymbol{Z}^\ell$

$$i - m \in [0, N_i^\ell - 1] \quad \text{and} \quad j - n \in [0, N_j^\ell - 1] . \tag{4.174}$$

When taking the derivative w.r.t. $Z_{i,j}^{\ell-1}$ only those factors where $m = m'$ and $n = n'$ are non-zero, hence

$$\frac{\partial Z_{i-m,j-n}^\ell}{\partial Z_{i,j}^{\ell-1}} = W_{m,n}^\ell f'(Z_{i,j}^{\ell-1}) \quad \text{or} \quad \frac{\partial Z_{i-m,j-n}^\ell}{\partial Z_{i,j}^{\ell-1}} = W_{m,n}^\ell f'(Z_{i,j}^{\ell-1}) , \tag{4.175}$$

for

$$i \in [0, N_i^{\ell-1} - 1], j =\in [0, N_j^{\ell-1} - 1], m \in [0, M_i^\ell - 1], n \in [0, M_j^\ell - 1] . \quad \text{or} \quad i \in \mathcal{N}^{\ell-1} \tag{4.176}$$

Otherwise, the Jacobian is zero.

We now shift the summation indices in (4.171) and obtain

$$\begin{aligned} \boldsymbol{E}^{\ell-1} &= \left[E_{i,j}^{\ell-1}\right] && (4.177)\\ &= \left[\sum_{m,n} E_{i-m,j-n}^{\ell} \frac{\partial Z_{i-m,i-n}^{\ell}}{\partial Z_{i,j}^{\ell-1}}\right] && (4.178)\\ &= \left[\left(\sum_{m,n} E_{i-m,j-n}^{\ell} W_{m,n}^{\ell}\right) f'(Z_{i,j}^{\ell-1})\right] && (4.179)\\ &= \left[\left(\sum_{k+m=i,l+n=j} E_{k,l}^{\ell} W_{m,n}^{\ell}\right) f'(Z_{i,j}^{\ell-1})\right] && (4.180)\\ &= f'(\boldsymbol{Z}^{\ell-1}) \circ (\boldsymbol{W}^{\ell} * \boldsymbol{E}^{\ell}) && (4.181) \end{aligned}$$

The convolution obviously is a full convolution.

Hence the size of the resulting matrix is, e.g., for the first index

$$N_i^{\ell} + M_i^{\ell} - 1 = (N_i^{\ell-1} - (M_i^{\ell} - 1)) + M_i^{\ell} - 1 = N_i^{\ell-1}\,, \tag{4.182}$$

as required, and consistent with the size of $f'(\boldsymbol{Z}^{\ell-1})$.

4.10.6 Synopsis

We collect the main relations for full and for convolutional connections between layers in Tab. 4.7. We observe the following intuitive relations, mainly resulting from the vectorization of the matrices when handling convolutional connection:

- The const functions are identical and related by vectorization.
- The recursive form of the forward propagation for full connections is a matrix multiplication, for convolutional connections a matrix correlation by construction. I.e. the weight matrices are to be flipped, i.e., mirrored at the centre, in order to realize convolutions.
- The L-weighted residuals are related by vectorization.
- The partial derivatives w.r.t. weights again are matrix multiplications and matrix correlations, respectively. I.e. the matrix valued ℓ-weighted residuals of convolutional connections need to be flipped.
- The partial derivatives w.r.t. biases are related by vectorization.
- Finally, the recursive structure of backward propagation for full connections contains a matrix vector multiplication of the weights with the ℓ-weighted residuals. For convolutional connections, we need a convolution of the weights and the ℓ- weighted residuals, – in contrast to the forward propagation, where we need to realize a correlation. In all cases we can use the correlation $\circledast$, when using the flip operation during backpropagation, as indicated in the last row.

4.10.7 Jacobian of the pooling

For being able to determine the gradient, we rewrite this operation in another form. We use the 2×2 submatrix

$$\boldsymbol{A}_{i,j}^{\ell} = \begin{bmatrix} A_{2i,2j}^{\ell} & A_{2i-1,2j}^{\ell} \\ A_{2i,2j-1}^{\ell} & A_{2i-1,2j-1}^{\ell} \end{bmatrix}\,, \tag{4.183}$$

referring to the index (i,j) of the output image, and the 4-unit vector $\boldsymbol{e}_{\alpha,i,j}^{\ell}$ denoting the maximum in this 2×2-matrix:

$$\boldsymbol{e}_{\alpha,i,j}^{\ell} = \left(\mathrm{vec}\boldsymbol{A}_{i,j}^{\ell} == \max\left(\mathrm{vec}\boldsymbol{A}_{i,j}^{\ell}\right)\right)\,. \tag{4.184}$$

		full	convolutional
cost function $\sum_n C_n/(2N)$	C_n	$\|\|\boldsymbol{a}_n^L - \boldsymbol{y}_n\|\|^2$	$\|\|A_n^L - Y_n\|\|^2$
forward propagation	recursion	$\boldsymbol{z}^\ell = W^\ell f(\boldsymbol{z}^{\ell-1}) + \boldsymbol{b}^\ell$	$Z^\ell = W^\ell \circledast f(Z^{\ell-1}) + B^\ell$
L-weighted residuals	$\boldsymbol{\varepsilon}^L, E_n^L$	$f'(\boldsymbol{z}^L) \circ (\boldsymbol{a}^L - \boldsymbol{y})$	$f'(Z^L) \circ (A^L - Y)$
partials w.r.t. weights	$\partial C/\partial W^\ell$	$\boldsymbol{\varepsilon}^\ell \boldsymbol{a}^{\ell-1,\mathsf{T}}$	$E^\ell \circledast A^{\ell-1}$
partails w.r.t. bias	$\partial C/\partial \boldsymbol{b}^\ell, \partial C/\partial B^\ell$	$\boldsymbol{\varepsilon}^\ell$	E^ℓ
backward propagation	recursion	$\boldsymbol{\varepsilon}^\ell = f'(\boldsymbol{z}^\ell) \circ \left(W^{\ell+1,\mathsf{T}} \boldsymbol{\varepsilon}^{\ell+1}\right)$	$E^\ell = f'(Z^\ell) \circ (W^{\ell+1} * E^{\ell+1})$ $E^\ell = f'(Z^\ell) \circ (\text{flip}\left(W^{\ell+1}\right) \overline{\circledast} E^{\ell+1})$

Table 4.7: Relations for deep networks. Full network for vectors, convolutional network for matrices. Transition from convolutional to full network by vectorization. Observe: in all cases we can use the correlation $\circledast$, as shown in the last row, there however being a full correlation

to arrive at the (i,j)-element of the pooled matrix $A^{\ell+1}$, which is reduced by a factor 2:

$$A_{i,j}^{\ell+1} = \boldsymbol{e}_{\alpha,i,j}^{\ell,\mathsf{T}} \text{vec} A_{i,j}^\ell \,. \tag{4.185}$$

Hence, the Jacobians for the backpropagation are the 4-unit vectors indicating which elements had been maximum during forward propagation:

$$\frac{\partial \text{vec} A_{i,j}^\ell}{\partial A_{i,j}^{\ell+1}} = \underset{4\times 1}{\boldsymbol{e}_{\alpha,i,j}^\ell} \,. \tag{4.186}$$

These vectors need to be stored during forward propagation.

5 Bayes- and Maximum-Likelihood-Estimation

The principle of Bayesian estimation and Maximum likelihood estimates is explained and illustrated for the simple example of observing an unknown entity for the cases with and without outliers. It is generalized to the linear Gauss-Markov model, indicating the close relation between generalized least squares estimation, ML-estimation and Bayesian estimation. The relation to the iteration scheme of Levenberg-Marquardt are given.

5.1 Preface

This note (2021) served as background for a lecture on the application of estimation theory to point cloud registration. It can be seen as a comment on the corresponding subsections (4.1.1 and 4.1.2) in Förstner and Wrobel (2016).

5.2 Goal

The goal of this note is to explain the principle of Bayesian estimation and of maximum-likelihood estimation. In detail, we want to formalize a set of uncertain statements using probability theory:

- We can model *preliminary information* or one's belief *belief* about unknown values of parameters, before performing any measurement of these parameters as *a priori* probability density.
- We can model the measuring process[1] and its uncertainty (or possible variations of the outcome of some observation) as conditional probability density. It may include the assumption that observations may be outliers. It may at the same time be used to characterize the certainty (*likelihood*) how a given observational value might explain the parameter.

[1] We distinguish between the measurement process and the measurement. The measurement process characterizes the physical boundary conditions for making the measurements, which includes the measuring instrument, the meteorological boundary conditions, the ability of the operator to handle the instrument, etc. The measurement is the result of using the measuring device and usually is a physical entity with its unit. We also call measurements observations, and the measurement process the observation process.

- We can fuse the preliminary information and the knowledge about the measurement process and describe it as conditional probability density, i.e., the *a posteriori* density , and in this manner model an *improved belief* about the parameters after performing the measurement process.
- From the model of the observation process and the available observations one can derive an optimal estimator[2] for the parameters, what is called the *maximum likelihood estimation* or ML estimation. In case the observation model allows for outliers, we arrive at robust estimators.
- If prior information is available, one can come to an optimal estimator for the parameters from the combination of this prior information, the observation model and the observations, the so-called *Maximum-a-posteriori-estimator*, or *MAP-estimator*. Since the so-called Bayes theorem is used, it is also called *Bayes estimation*.
- The Bayesian estimator can serve as an explanation for an estimation method proposed by Levenberg and Marquardt, which also works in the case a parameter vector is not completely observed.

We illustrate the connections using the determination of a distance and the classic Gauss–Markov model.

5.3 Establishing the model

We refer to the following two examples

1. First, we handle the simple case of the length θ of a box, which we want to determine by some measurement. For our purpose, we assume that we have some preliminary knowledge about the length and wish to combine it with the knowledge about the measuring process and the concrete result of a measurement. Hence, we want to correct our belief, we have before the actual measurement, using the result of the measurement process. The goal is to describe this situation probabilistically and use it for some best estimator for the length of the box.
2. We then generalize the situation for the case of the linear Gauss–Markov model.

5.3.1 The preliminary knowledge

In our context we start from the following assumptions: We assume the length of the box is 1.2 dm or 1.6 dm, since we are not sure which of two boxes we have. However, we presume we are more likely to measure the shorte side. The preliminary knowledge about the length is uncertain by 0.1 dm.

This preliminary knowledge we may formalize in the following manner: With a probability $P(\mu = 1.2)$ the length is normally distributed with mean $\mu_1 = 1.2$ dm and standard deviation $\sigma_1 = 0.1$ dm. With the same probability $P(\mu = 1.6$ the length of has a mean of $\mu_2 = 1.6$ dm and the same standard deviation $\sigma_2 = 0.1$ dm.

This preliminary knowledge can be visualized as probability density, as in the following 5.1 The area below the curve of course is 1, and both densities around 1.2 dm and around 1.6 dm contribute the same way. Using the Gaussian density function

$$g(\theta \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{(\theta - \mu)}{\sigma}\right)^2\right) \tag{5.1}$$

the preliminary knowledge would formally be

$$\underline{\theta} \sim p(\theta) = P(\mu = 1.2)g(\theta \mid 1.2, 0.1^2) + P(\mu = 1.6)g(\theta \mid 1.6, 0.1^2)\,. \tag{5.2}$$

[2] An estimator is a rule for deriving an estimate of a parameter (vector)

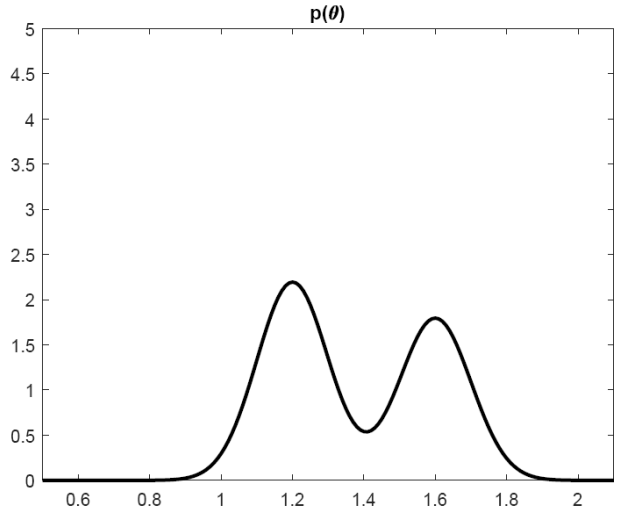


Figure 5.1: Preliminary knowledge about the parameter θ represented as probability density, We assume, that the smaller mean is a bit more probable, namely $P(\mu = 1.2) = 0.55$ and thus $P(\mu = 1.6) = 0.45$

The equation means the following: The random variable $\underline{\theta}$ represents our uncertain knowledge about the length. It follows the distribution shown by the density $p(\theta)$.

In statistical nomenclature there are two ways to express this situation:

1. The *prior distribution*[3] of $\underline{\theta}$ is characterized by the density $p(\theta)$.
2. The *belief*, that we have about the length $\underline{\theta}$ prior to some measurement is characterized by $p(\theta)$.

The second formulation has the advantage, that we make explicit, that our belief may, possibly stepwise, change by future measurements.

5.3.2 The measurement process

Now we model the measuring process. The model describes the uncertain relation between the observations and the parameter. The model provides two types of information:

1. How large are the deviations between the observation and the parameter. This is equivalent to describe the uncertainty of the measurements, in case the parameter value is known.
2. How well does the observation explain properties of the parameter? This is equivalent to the question: How certain are we about the value of the of the parameter if we know the observational value?

5.3.2.1 The measurement

Let us assume, the measurement device has an uncertainty of 0.15 dm. I.e. the deviations of the observation and the length of the box are 0.15 dm on an average. Again, we formalize this statement using a density function, namely the conditional density $p(y \mid \theta)$, which tells how the random variable $\underline{y}$ is distributed in case we know θ:

$$\underline{y} \mid \theta \sim p(y \mid \theta) = g(l \mid \theta, \sigma^2) \propto \exp\left(-\frac{1}{2}\left(\frac{y-\theta}{\sigma}\right)^2\right). \tag{5.3}$$

The vertical bar "|" indicates, that we have full knowledge about what stands right of "|" which we take as fixed information.

[3] Random variables are underscored

5.3.2.2 Explanation of the parameters by the observation – the likelihood

We may interpret $p(y \mid \theta)$ in an alternative manner, since it only depends on the difference $y - \theta$, see (5.3): In case the measurement value y in known, the function $p(y \mid \theta)$ tells us in some way how well we may explain y by the parameters.

The English language has several words for expressing uncertain or uncertainty: `probable` or `probability` and `likely` or `likelihood`. [4]

Therefore, the function $p(y \mid \theta)$ for fixed y also is called `likelihood` or `likelihood function`:

$$L(\theta) = p(y \mid \theta)\,, \tag{5.4}$$

where one implicitly assumes $L(\theta)$ refers to a specific observation y, sometimes added as index, thus $L_y(\theta)$. Observe, the function $L(\theta)$ is no density, since the integral over θ is not 1 in general.

If in our example we observe $y = 1.5$ dm, and assume the measurement process postulates an uncertainty of $\sigma = 0.15$ dm then we obtain for this observation the likelihood

$$L(\theta) = p(1.5 \mid x) = g(1.5 \mid x, 0.15^2) \propto \exp\left(-\frac{1}{2}\left(\frac{1.5 - x}{0.15}\right)^2\right)\,. \tag{5.5}$$

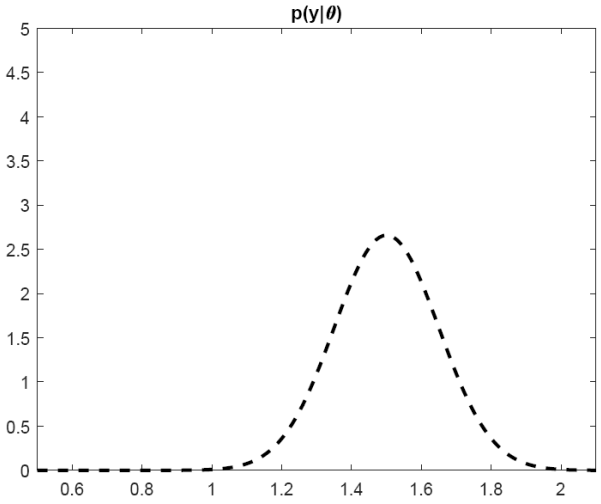


Figure 5.2: Likelihood-function $L(\theta) = p(y \mid \theta)$ for θ, if the uncertain observation is $y = 1.5$ and has standard deviation 0.15 dm

5.3.3 The fusion of preliminary knowledge and measurement

Now we fuse these two types of information and interpret its result in two ways.

5.3.3.1 Bayes' rule

Scalar valued form of Bayes' rule. We assume that the preliminary knowledge has no influence onto the measurement. Then we have two mutually independent statements about θ represented by their densities $p(\theta)$ and $p(y \mid \theta)$.

[4] The German language only has one word for `probable` (wahrscheinlich) or `probability` (Wahrscheinlichkeit). If we use the German word `Sicherheit` (certainty) it implies a high probability, since otherwise we use the word `Unsicherheit`. Statistical nomenclature distinguishes the two: The word `likelihood` is used, in case we want to express the `certainty/uncertainty` of an event, but one wants to avoid the word `probability`, since generally the `likelihood` does not have the mathematical properties of a probability or density, e.g., it does not sum/integrate to 1.

The statement `I know that` θ `is distributed according` $p(\theta)$ `and (simultaneously) that the uncertainty of the measurement is characterized by` $p(y \mid \theta)$ therefore has a density which is the product of the two densities of concern:

$$p(\theta)\, p(y \mid \theta)\,. \tag{5.6}$$

But this is identical with the joint density of y and θ

$$p(\theta \mid y) = p(\theta)\, p(y \mid \theta)\,. \tag{5.7}$$

This density is difficult to interpret. We really would like to known the most probable value for θ in case we know the observation y or we want to characterize the belief about θ after having observed it. This belief can be characterized by the conditional density $p(\theta \mid y)$, whose maximum may be taken as a plausible estimator for θ.

To achieve this goal, we apply the relation $p(\theta, y) = p(\theta)\, p(y \mid \theta)$ in a second form, where y and θ are exchanged and in a first step obtain

$$p(\theta, y) = p(\theta)\, p(y \mid \theta) = p(y)\, p(\theta \mid y)\,. \tag{5.8}$$

Solving for $p(\theta \mid y)$ directly yields

$$\boxed{p(\theta \mid y) = \frac{p(\theta)\, p(y \mid \theta)}{p(y)}\,.} \tag{5.9}$$

This is the *a posteriori density* of $\underline{x}$ for a given observational value y. For our example, the function is shown in the Fig. 5.3 together with $p(\theta)$ and $p(y \mid \theta)$.

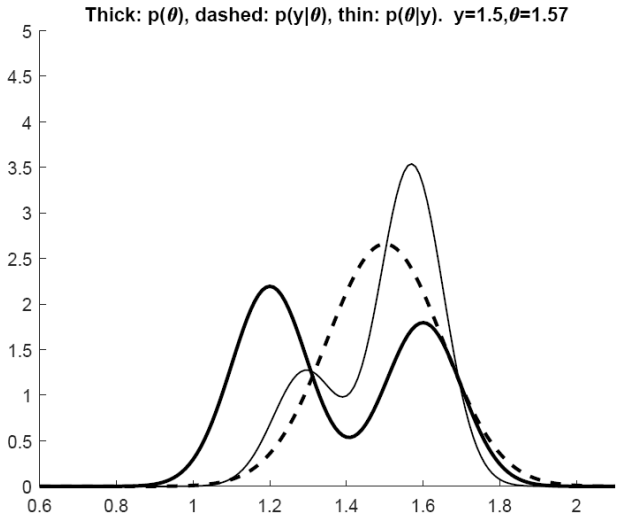


Figure 5.3: The prior density $p(\theta)$ (bold) shows the belief, that the length θ of the box is around 1.2 dm or 1.6 dm. The likelihood function $L(\theta) = p(y \mid \theta)$ (dashed) results from the measurement $y = 1.5$ The joint density $p(\theta, y)$ (thin) is proportional to the posterior density $p(\theta \mid y)$, since we have $p(\theta, y) = p(\theta \mid y)p(y)$ and the density $p(y)$ is constant, since the observational value l is a fixed entity. the conditional density $p(\theta \mid y)$ characterized the uncertainty of our belief about θ after knowing the value of the observation. We can take as best estimator the maximum value, which is called the Bayes-estimator or a posterior estimator. This integrates the preliminary knowledge and our measurement value. The maximum of $p(\theta \mid y)$ is close to $\widehat{\theta}_{\text{Bayes}} \approx 1.57$, closer to the right local maximum of the a priori density. In case we do not use the preliminary knowledge, but only the knowledge about the measurement process, i.e., the likelihood-function, then the best estimator can be taken as the maximum of $p(y \mid \theta)$ called the maximum likelihood estimator $\widehat{x}_{\text{ML}} = 1.5$ which in our simple case is identical to the measurement value

Obviously, the uncertainty about the length θ of the box is diminished, since the peak at 1.57 dm is narrower than that at the observation of the preliminary values. Instead of two relative maxima of the same height we now have two maxima of clearly different height. The absolute optimum is close to 1.57 dm. It lies between the observational value and the right maximum of the prior density. However, we might not be sure, that it is the best choice to take this absolute maximum as best value for θ. One might be tempted to report about the second (largest) relative maximum, or – instead of the maximum value take the mean value of $p(\theta \mid y)$ - alternatives we do not discuss here further.

Robust likelihood. Generally we cannot assume, that the belief is supported by observations. In case preliminary knowledge and observations collide, e.g., if we would have measured $y = 2.2$ dm, the belief about the length of the box would have been diminished, approximately by a factor $1/2$ – unless we would have allowed outliers within the measuring process or in case we would have allowed to update our model for the measuring process, e.g., by concluding that the uncertainty of the observations may be adapted, e.g., including the variance of the measurements into the parameter vector.[5]

We now show the power of this setup and assume that the measurment my be an outlier. This directly has an influence onto the likelihood: We need to specify our belief, say P_{out}, that an measurement is an outlier and in which range it lies, say uniformly in the range $[a, b]$. The the prior could be specified by

$$p(y \mid \theta, \texttt{possibleoutlier}) = (1 - P_{\text{out}})p(\theta \mid \texttt{nooutlier}) + P_{\text{out}}p(\theta \mid \texttt{outlier}) \,. \quad (5.10)$$

This leads to a *robust likelihood*, which allows observations – practically – not to influence the estimate.

In the Fig. 5.4 we assume $P_{\text{out}} = 0.2$ and $[a, b] = [0, 6]$. Hence, the prior would be

$$p(y \mid \theta, \texttt{possibleoutlier}) = 0.8g(1.5 \mid \theta, 0.15^2) + 0.2\mathcal{U}(0, 6) \,. \quad (5.11)$$

The combination of the prior and the likelihood function leads to a quite different result

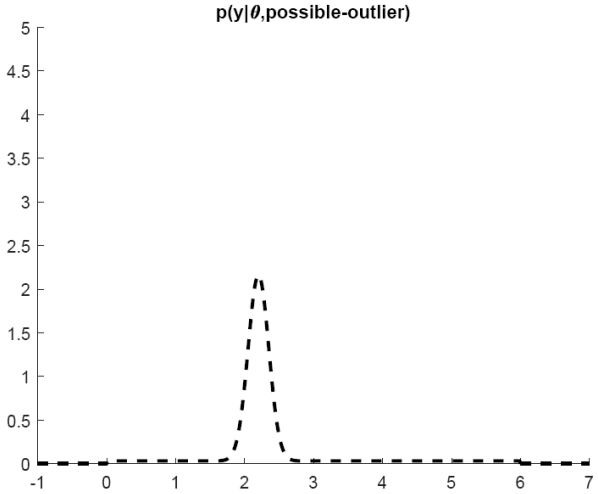


Figure 5.4: Likelihood-function $L(\theta) = p(y \mid \theta)$ for θ, if the uncertain observation is $y = 2.2$ and has standard deviation 0.15 dm, thus actually is an outlier. We assume with a probability of 20 % the measurment may be in the range between 0 and 6. The density in the interval $[0, 6]$ therefore is larger than $0.2/6 = 1/30$. The density outside the interval is practically 0

than before. An estimator based on a likelihood which allows for outliers usually is termed a robust estimator, as in this example.

[5] It is interesting to think about the meaning of the above reasoning in case we replace the notion `belief` by the notion `prejudice`.

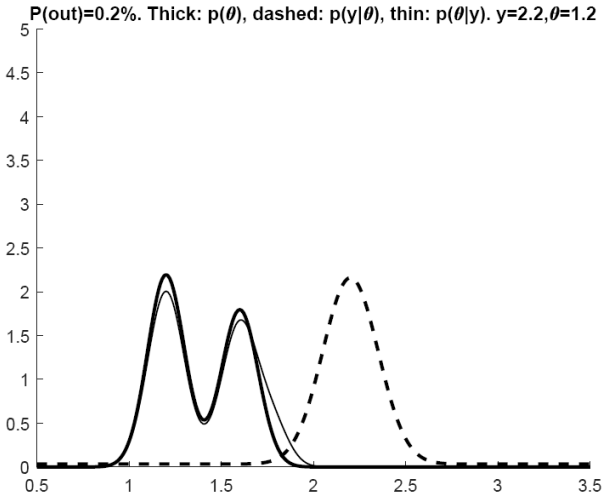


Figure 5.5: A case with an outlier and robust likelihood. The prior density is the same as before. The density $p(\theta)$ (bold) shows the belief, that the length θ of the box is around 1.2 dm or 1.6 dm, and the ssmaller mean is a bit more likely. The likelihood function $L(\theta) = p(y \mid \theta, \texttt{possibleoutlier})$ (dashed) results from the measurement $y = 2.2$, which is an outlier, but we do not know this. However, we allow that observations may be outliers. This leads to a robust likelihood function $p(y \mid \theta, \texttt{possibleoutlier})$. The joint density $p(\theta, y, \texttt{possibleoutlier})$ (thin) is proportional to the posterior density $p(\theta \mid y, \texttt{possibleoutlier})$, since we have $p(\theta, y, \texttt{possibleoutlier}) = p(\theta \mid y, \texttt{possibleoutlier})p(y)$ and the density $p(y)$ is constant, since the observational value y is a fixed entity. The conditional density $p(\theta \mid y, \texttt{possibleoutlier})$ characterized the uncertainty of our belief about θ after knowing the value of the observation. We again can take as best estimator the maximum value. The maximum of $p(\theta \mid y, \texttt{possibleoutlier})$ is close to $\widehat{\theta}_{\text{Bayes}} \approx 1.2$ very close to the left local maximum of the a priori density, since the likelihood alows the observation to be an oulier and – practically – not influence the prior. But since the prior density for the smaller mean $\mu = 1.2$ is higher than for the larger mean $\mu = 1.6$, the best estimate is close to the smaller mean. The prior in the range of the observation around $y = 2.2$ is very small, compared with the likelihood, except for a small range between 1.7 and 2.0, where the prior and the observation compete

Vector valued form of Bayes' rule. Eq. (5.9) is called *Bayes equation* or *Bayes-Theorem* and is central to statistical pattern recognition. In our context we use it as basis for fusion of prior knowledge and observations. The eq. also holds for vector (representing sets) of observations and parameters, $\boldsymbol{y}$ and $\boldsymbol{\theta}$:

$$\boxed{p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{\theta})\, p(\boldsymbol{y} \mid \boldsymbol{\theta})}{p(\boldsymbol{y})}\,.} \tag{5.12}$$

In our context of parameter estimation from observations it contains:

1. The *a priori density* $p(\boldsymbol{\theta})$ for the parameters, which we want to determine. This prior information may come from rough estimates, from maps, from earlier measurements, and so on. It specifies the belief we have about $\boldsymbol{\theta}$ before performing the measurement process leading to measurements $\boldsymbol{y}$.
2. The conditional density $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ characterizes the uncertainty we have about the measuring process. It is the (conditional) density of the observations in case the parameters were known, thus in essence model the observations as a function of the parameters. from their predictions and their random deviations during the measurement process. In case it is interpreted as the likelihood-function $L(\boldsymbol{\theta}) := p(\boldsymbol{y} \mid \boldsymbol{\theta})$, it tells how well the parameters explained the observations.

3. The conditional density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ is the a posterior density of the parameters. It characterizes the uncertainty of the parameters $\boldsymbol{\theta}$, after the observational values $\boldsymbol{y}$ ar known. It also specifies the belief we have about $\boldsymbol{\theta}$ corrected by the measurements.
4. The density $p(\boldsymbol{y})$ is a constant if the measurement values are given. It also can be interpreted as factor for enforcing the integral of the right-hand side over $\boldsymbol{\theta}$ to be 1.

5.3.3.2 Belief propagation, correction of belief

The belief we have about $\boldsymbol{\theta}$ occurs twice in the Bayes equation. We want to denote it with $\mathrm{bel}_t(\boldsymbol{\theta})$, where the index indicates the situation or time the belief refers to.

1. The belief about $\boldsymbol{\theta}$ before measuring e.g., is
$$\mathrm{bel}_{\text{a priori}}(\boldsymbol{\theta}) := p(\boldsymbol{\theta})\,. \tag{5.13}$$
2. The belief about $\boldsymbol{\theta}$ afte the measurement then is
$$\mathrm{bel}_{\text{a posteriori}}(\boldsymbol{\theta}) := p(\boldsymbol{\theta} \mid \boldsymbol{y})\,. \tag{5.14}$$

They are connected by the likelihood $L(\boldsymbol{\theta}) := p(\boldsymbol{y} \mid \boldsymbol{\theta})$ which tells how well the observations are explained by $\boldsymbol{\theta}$

$$\mathrm{bel}_{\text{a posteriori}}(\boldsymbol{\theta}) = k\, p(\boldsymbol{y} \mid \boldsymbol{\theta})\, \mathrm{bel}_{\text{a priori}}(\boldsymbol{\theta}) \quad \text{with} \quad k = \frac{1}{p(y)}\,. \tag{5.15}$$

In case we imagine a situation where at different times we perform observations we also could write

$$\mathrm{bel}_t(\boldsymbol{\theta}) = k_t\, p(\boldsymbol{y}_t \mid \boldsymbol{\theta})\, \mathrm{bel}(\boldsymbol{\theta}_{t-1}) \quad \text{with} \quad k_t = \frac{1}{p(\boldsymbol{y}_t)}\,. \tag{5.16}$$

The constant value k_t serves for normalization, since beliefs in our context are conditional densities. The index t of the observations $\boldsymbol{y}_t$ refers to the time, when the observations have been mad, and in case it refers to the parameters, the time before and after performing the measurement. as shown in the diagram Fig. 5.6 with the time axis running from left to right.

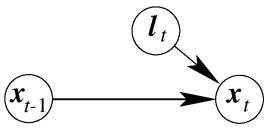


Figure 5.6: Belief propagation, correction of the belief about $\boldsymbol{\theta}$ caused by observations $\boldsymbol{y}$

5.4 Estimation principles

There are many principles to motivate an estimation. We present two of them, one which is based on the likelihood-function $L(\boldsymbol{\theta})$ and one which is based on the a posteriori density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$.

5.4.1 The maximum-likelihood-estimation and its generalization

In case we only know the measuring process and the actual observations, it is plausible to choose as best value for the parameter θ the one which optimally explains the observed value, i.e., the one where the likelihood-function $L(\theta)$ is maximal. The we obtain, what is called the *Maximum-Likelihood estimation* (ML estimation) from

$$\widehat{x}_{\mathrm{ML}} = \mathrm{argmax}_x\, L(\theta) = \mathrm{argmax}_x\, p(y \mid \theta)\,. \tag{5.17}$$

Actually, we used this argumentation as intuitively plausible, when we concluded that the observed value $k = 1.5$ dm is a good estimate for θ.

We easily can generalize this principle to multiple observation and parameters: In case we have an unknown U-vektor $\boldsymbol{\theta}$ which we want to derive from a set of N observations $\boldsymbol{y}$ and are able to characterize the measuring process probabilistically, i.e., are able to provide the density of the observations for given parameters

$$\underline{\boldsymbol{y}} \mid \boldsymbol{\theta} \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}) \tag{5.18}$$

then together with the likelihood-function

$$L(\boldsymbol{\theta}) = p(\boldsymbol{y} \mid \boldsymbol{\theta}) \tag{5.19}$$

we obtain the ML-estimator from

$$\boxed{\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \mathrm{argmax}_x \; L(\boldsymbol{\theta}) = \mathrm{argmax}_x \; p(\boldsymbol{y} \mid \boldsymbol{\theta})\,.} \tag{5.20}$$

The principle is very general. Take as second example the Gauss–Markov model in the form

$$\underline{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\theta} + \underline{\boldsymbol{e}} \quad \text{und} \quad \underline{\boldsymbol{e}} \sim p(\boldsymbol{e}) = g(\boldsymbol{0}, \Sigma_{ee})\,. \tag{5.21}$$

here the design matrix is $\boldsymbol{Z}$ and the observational deviations $\underline{\boldsymbol{e}}$ have mean $\boldsymbol{0}$ and covariance matrix Σ_{ee}. We rewrite this model and characterize the observations as a function of the parameters in a probabilistic manner, we may write

$$\underline{\boldsymbol{y}} \mid \boldsymbol{\theta} \sim g(\mathbb{E}(\underline{\boldsymbol{y}} \mid \boldsymbol{\theta}), \mathbb{D}(\underline{\boldsymbol{y}} \mid \boldsymbol{\theta})) \quad \text{mit} \quad \mathbb{E}(\underline{\boldsymbol{y}} \mid \boldsymbol{\theta}) = \boldsymbol{X}\boldsymbol{\theta} \quad \text{und} \quad \mathbb{D}(\underline{\boldsymbol{y}} \mid \boldsymbol{\theta}) = \Sigma_{ee} \tag{5.22}$$

The left-hand side shows the random vector "$\underline{\boldsymbol{y}}$ for given $\boldsymbol{\theta}$" whose distribution is related to the situation, which is unknown up to the parameter vector $\boldsymbol{\theta}$. The we also can write

$$\underline{\boldsymbol{y}} \mid \boldsymbol{\theta} \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}) = g(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \Sigma_{ee})\,. \tag{5.23}$$

or explicitely

$$\underline{\boldsymbol{y}} \mid \boldsymbol{\theta} \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \frac{1}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^{\mathsf{T}}\Sigma_{ee}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})\right)\,. \tag{5.24}$$

The ML estimator results from maximizing $p(\boldsymbol{y} \mid \boldsymbol{\theta})$. Since the exponential function is monotonically increasing, we also can obtain the ML estimator by minimize the negative logarithm $-\log p(\boldsymbol{y} \mid \boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \mathrm{argmin}_x \; (-\log L(\boldsymbol{\theta})) = \mathrm{argmin}_x \; (-\log p(\boldsymbol{y} \mid \boldsymbol{\theta}))\,. \tag{5.25}$$

For the case of a linear Gauss–Markov model and omitting constant terms we then obtain

$$\boxed{\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \mathrm{argmin}_x \; (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^{\mathsf{T}}\Sigma_{ee}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})\,.} \tag{5.26}$$

This obviously is a weighted least squares estimation[6], using the invers covariance matrix as weight matrix,

$$\boxed{\hat{\boldsymbol{\theta}}_{\mathrm{KQ}} = \mathrm{argmin}_x \; (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) \quad \text{mit} \quad \boldsymbol{W} := \Sigma_{ee}^{-1}\,,} \tag{5.27}$$

with the solution based on the normal equation system

$$\boldsymbol{X}^{\mathsf{T}}\Sigma_{yy}\boldsymbol{X}\;\hat{\boldsymbol{\theta}}_{ML} + \boldsymbol{X}^{\mathsf{T}}\Sigma_{yy}\boldsymbol{y} = \boldsymbol{0}\,. \tag{5.28}$$

If we argue backward, we may interpret each weighted least squares estimator as ML estimator by saying: The weighted least squares estimator – implicitly, i.e., without assuming this explicitly or without mentioning – assumes as stochastical model a Gaussian distribution for with mean zero and covariance matrix $\Sigma_{ee} = \boldsymbol{W}^{-1}$. This allows to check the plausibility of the weighted least squares estimator, especially the plausibility of the assumed weight matrix.

[6] Sometimes called generalized least squares (GLS)

5.4.2 The Bayes or maximum a posterior estimator

Now we, in addition to the observations, also want to use some preliminary information about the parameters. Then we obtain the following estimator from the posteriori density:

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MAP}} = \hat{\boldsymbol{\theta}}_{\text{Bayes}} = \text{argmax}_x \; p(\boldsymbol{\theta} \mid \boldsymbol{y})\,.} \tag{5.29}$$

It si the *maximum a posteriori estimator* for $\boldsymbol{\theta}$ or MAP-estimator. Since the a posterior density is kinked with the a priory density via the Bayes equation, this estimator sometimes is called the *Bayes estimator* of $\boldsymbol{\theta}$. [7]

As we know for the Bayes equation (5.12) the denominator is irrelevant, since it is a constant for given observational values. Therefore we also may define the Bayes estimator as

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MAP}} = \hat{\boldsymbol{\theta}}_{\text{Bayes}} = \text{argmax}_x \; p(\boldsymbol{\theta}) p(\boldsymbol{y} \mid \boldsymbol{\theta})\,.} \tag{5.30}$$

If we now take the Gauss–Markov model and assume we have some prior information about the parameters, e.g., when estimating coordinates and taking approximate values $\boldsymbol{\theta}_0$ of the point coordinates from a map, and in the most simple case represent this uncertain prior information as normal distribution with mean $\boldsymbol{\theta}_0$ and some covariance matrix $\Sigma_{x_0x_0}$. The we have two types of "observations": (1) the – possibly quite vague – prior information $\{\boldsymbol{\theta}_0, \Sigma_{x_0x_0}\}$ and (2) the measurement values following a measurement process ($\{\boldsymbol{y}, \Sigma_{yy}\}$), now using $\Sigma_{ee} = \Sigma_{yy}$:

$$\underline{\boldsymbol{\theta}}_0 \quad \sim p(\boldsymbol{\theta}) \quad = g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_0, \Sigma_{x_0x_0}) \tag{5.31}$$

$$\underline{\boldsymbol{y}} \mid \boldsymbol{\theta} \quad \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}) \quad = g(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \Sigma_{yy}) \tag{5.32}$$

One can show, the product (5.30) of the two normal distributions (5.31) und (5.32) can be rearranged and the Bayes estimator is identical with the ML estimator, in case we take both observational groups for determining the parameters, see Bishop (2006, Sect. 2.3.2). Therefore, the model for estimating the parameters with prior information can be written as

$$\begin{bmatrix} \underline{\boldsymbol{\theta}}_0 \\ \underline{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{X} \end{bmatrix} \boldsymbol{\theta} + \begin{bmatrix} \underline{\boldsymbol{e}}_{x_0} \\ \underline{\boldsymbol{e}} \end{bmatrix} \quad \text{mit} \quad \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{e}}_{x_0} \\ \underline{\boldsymbol{e}} \end{bmatrix} \right) = \begin{bmatrix} \Sigma_{x_0x_0} & \boldsymbol{0} \\ \boldsymbol{0} & \Sigma_{yy} \end{bmatrix}. \tag{5.33}$$

This leads to the following estimator

$$\hat{\boldsymbol{\theta}}_{\text{Bayes}} = (\Sigma^{-1}_{x_0x_0} + \boldsymbol{X}^{\mathsf{T}}\Sigma_{yy}\boldsymbol{X})^{-1}(\Sigma^{-1}_{x_0x_0}\boldsymbol{\theta}_0 + \boldsymbol{X}^{\mathsf{T}}\Sigma_{yy}\boldsymbol{y})\,. \tag{5.34}$$

This is identical to

$$\hat{\boldsymbol{\theta}}_{\text{Bayes}} = (\Sigma^{-1}_{x_0x_0} + \Sigma^{-1}_{\hat{\theta}\hat{\theta},\text{ML}})^{-1}(\Sigma^{-1}_{x_0x_0}\boldsymbol{\theta}_0 + \Sigma^{-1}_{\hat{\theta}\hat{\theta},\text{ML}}\hat{\boldsymbol{\theta}}_{\text{ML}})\,. \tag{5.35}$$

Thus we obtain the important insight: *Die Bayes estimator, $\hat{\boldsymbol{\theta}}_{Bayes}$, is the* weighted mean *of the prior information $\{\boldsymbol{\theta}_0, \Sigma_{\theta_0\theta_0}\}$ and the result $\{\hat{\boldsymbol{\theta}}_{ML}, \Sigma_{\hat{\theta}\hat{\theta},ML}\}$ of the ML estimator (without prior information), when taking the inverse covariance matrices as weights.*

Finally, we establish a relation between the two estimates. We may write the Bayes equation also as

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{\theta})}{p(\boldsymbol{y})}\, p(\boldsymbol{y} \mid \boldsymbol{\theta}) \tag{5.36}$$

[7] In the statistical literature sometimes the terms `ML estimator` and `Bayes estimator` are differentiated: In case only probabilistic information is used, thus the a posteriori density is optimized, the estimator is called `MAP estimator`. If in addition some expected costs for the estimated value are taken into account, e.g., by its deviation from the mean value, the estimator is called `Bayes estimator`, reflecting the motivation of Thomas Bayes (1701–1761), who was interested in optimizing his games' stake. Here, we use the two terms interchangeable.

In case the density $p(\boldsymbol{\theta})$ of the prior information is a constant, this is equivalent to the statement: We do not have any prior information. [8] The both conditional densities are proportional.

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = k\, p(\boldsymbol{y} \mid \boldsymbol{\theta})\,. \tag{5.37}$$

The we have

$$\widehat{\boldsymbol{\theta}}_{\text{Bayes}} = \operatorname{argmax}_x p(\boldsymbol{\theta} \mid \boldsymbol{y}) \equiv \widehat{\boldsymbol{\theta}}_{\text{ML}} = \operatorname{argmax}_x p(\boldsymbol{y} \mid \boldsymbol{\theta})\,. \tag{5.38}$$

The maximum likelihood estimator can be interpreted as Bayes estimate without prior.

For the Gauss–Markov model we observe, that the right hand side on (5.24) – upto a constant factor – is identical to $p(\boldsymbol{\theta} \mid \boldsymbol{y})$. Hence, we have a relation between the weighted least squares estimator and the Bayes estimator:

The weighted least squares estimator for a linear model and Gaussian distribution of the observational deviations is identical with the Bayes estimator without prior information.

5.5 Incomplete measurements and Levenberg-Marquardt method

The Bayesian estimator allows to determine parameters even in case we do not have enough observations to estimate them using the ML estimator. Then the corresponding normal equation matrix, the matrix $X^{\mathsf{T}}\Sigma_{yy}^{-1}X$ in (5.28), is singular:

- If the observations do not allow we still can estimate the parameters following (5.35)

  $$\widehat{\boldsymbol{\theta}}_{\text{Bayes}} = (\Sigma_{x_0x_0}^{-1} + X^{\mathsf{T}}\Sigma_{yy}^{-1}X)^{-1}(\Sigma_{x_0x_0}^{-1}\boldsymbol{\theta}_0 + X^{\mathsf{T}}\Sigma_{yy}^{-1}\boldsymbol{y})\,. \tag{5.39}$$

  As an example, in case a scene point, say, with coordinates $\boldsymbol{\theta}_{0k}$, is not observed, then in spite of the singularity of the resulting normal equation matrix, we obtain a solution, reproducing the a priori values $\boldsymbol{\theta}_{0k}$ of the coordinates.

- Similarly, in case we do not know, whether all parameters are estimable, we might introduce prior values for the parameters in order to guarantee estimability within a Bayes estimation. This is easy, especially also within an iterative procedure (assuming the model is nonlinear), since we may set the prior for the corrections $\Delta\boldsymbol{\theta}_0$ to zero and simply choose a multiple of a unit matrix as covariance matrix $\Sigma_{x_0x_0}$. Then we obtain the corrections to the parameters as

  $$\widehat{\Delta\boldsymbol{\theta}}_{\text{LM}} = (\lambda I + X^{\mathsf{T}}\Sigma_{yy}^{-1}X)^{-1}\; X^{\mathsf{T}}\Sigma_{yy}^{-1}\Delta\boldsymbol{y}\,. \tag{5.40}$$

  Levenberg (1944) and Marquardt (1963) have proposed to diminish the parameter λ, which actually is a common weight for the prior values $\Delta\boldsymbol{\theta}_0$, during the sequence of iterations ν, say choosing an exponential decay $\lambda^{(\nu)} = \lambda_0^{\nu}$, such that the prior information on one hand allows an estimation on the other hand does not disturb the solution.

5.6 Appendix: Probability $p(y)$ for the double mixture model

The posterior density is given by

$$\mathrm{p}(\theta \mid y) = \frac{\mathrm{p}(y \mid \theta)\mathrm{p}(\theta)}{\mathrm{p}(y)}\,. \tag{5.41}$$

[8] In a finite universe the range of $\boldsymbol{\theta}$ is limited, such that $p(\boldsymbol{\theta}) = c > 0$. In case we restrict the range of the normal distribution accordingly, the effect onto the density is negligible.

For drawing the figures we determined $\mathrm{p}(y)$ numerically:

$$\mathrm{p}(y) \approx \sum_{\theta_i} \mathrm{p}(y \mid \theta_i)\mathrm{p}(\theta_i)\Delta\theta\,. \tag{5.42}$$

We mow want to give an explicit expression for the doubly mixture model with the prior

$$\mathrm{p}(\theta) = P(\mu = \mu_1)\ \mathrm{g}(\theta \mid \mu_1, w_1) + P(\mu = \mu_2)\ \mathrm{g}(\theta \mid \mu_2, w_2) \tag{5.43}$$

and the likelihood

$$L(\theta) = \mathrm{p}(y \mid \theta) = \mathrm{P_{in}}\ \mathrm{g}(y \mid \theta, w_y) + (1 - \mathrm{P}_{in})\ U(a, b)\,. \tag{5.44}$$

The denominator of the posterior is

$$\mathrm{p}(y) = \int_\theta \mathrm{p}(y \mid \theta)\mathrm{p}(\theta)\mathrm{d}\theta\,. \tag{5.45}$$

It is a sum of four integrals

$$\mathrm{p}(y) = A_1 + A_2 + B_1 + B_2 = \int_\theta (a_1 + a_2 + b_1 + b_2)\,\mathrm{d}\boldsymbol{\theta}\,. \tag{5.46}$$

of the following terms

$$\begin{aligned}
a_1 &= P(\mu = \mu_1)\mathrm{P_{in}}\ \ \mathrm{g}(\theta \mid \mu_1, w_1)\mathrm{g}(y \mid \theta, w) && (5.47)\\
a_2 &= P(\mu = \mu_2)\mathrm{P_{in}}\ \ \mathrm{g}(\theta \mid \mu_2, w_2)\mathrm{g}(y \mid \theta, w) && (5.48)\\
b_1 &= P(\mu = \mu_1)\ (1 - \mathrm{P_{in}})\ \mathrm{g}(\theta \mid \mu_1, w_1)U(a, b) && (5.49)\\
b_2 &= P(\mu = \mu_2)\ (1 - \mathrm{P_{in}})\ \mathrm{g}(\theta \mid \mu_2, w_1)U(a, b)\,. && (5.50)
\end{aligned}$$

The first two terms contain the factors of two Gaussians

$$a_{i0} = \frac{1}{2\pi\sigma_i\sigma_\theta} \exp\left(-\frac{1}{2}w_i(\theta - \mu_i)^2\right) \exp\left(-\frac{1}{2}w(\theta - y)^2\right)\,, \tag{5.51}$$

while the second two terms contain factors of a Gaussian with an uniform density

$$b_{i0} = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2}w_i(\theta - \mu_i)^2\right) U(a, b)\,. \tag{5.52}$$

We simplify the argument q_i of the exponential in a_{i0}. It is a sum of two quadratic forms

$$q_i = w_i(\theta - \mu_i)^2 + w(\theta - y)^2\,. \tag{5.53}$$

It can be rewritten as

$$\begin{aligned}
q_i &= (w_i + w)\ \theta^2 - 2(w_i\mu_i + wy)\ \theta + (w_i\mu_i^2 + wy^2) && (5.54)\\
&= (w_i + w)\left(\theta^2 - 2\frac{w_i\mu_i + wy}{w_i + w} + \frac{w_i\mu_i^2 + wy^2}{w_i + w}\right) && (5.55)\\
&= (w_i + w)\left(\left(\theta - \frac{w_i\mu_i + wy}{w_i + w}\right)^2 + \left(\frac{w_i\mu_i^2 + wy^2}{w_i + w} - \left(\frac{w_i\mu_i + wy}{w_i + w}\right)^2\right)\right) && (5.56)\\
&= (w_i + w)\left(\left(\theta - \frac{w_i\mu_i + wy}{w_i + w}\right)^2 + \frac{w\,w_i\,(\mu_i - y)^2}{(w + w_i)^2}\right) && (5.57)\\
&= \frac{\left(\theta - \frac{w_i\mu_i + wy}{w_i + w}\right)^2}{1/(w_i + w)} + \frac{w\,w_i\,(\mu_i - y)^2}{(w + w_i)} && (5.58)
\end{aligned}$$

We now take the weighted mean

$$\bar{\mu} = \frac{w_i\mu_i + wy}{w_i + w} \tag{5.59}$$

the corresponding variance

$$\bar{\sigma} = \frac{1}{w_i + w} = \frac{1}{\frac{1}{\sigma_i^2} + \frac{1}{\sigma^2}} = \frac{\sigma_i^2 \sigma^2}{\sigma_i^2 + \sigma^2} \tag{5.60}$$

and the weigthted squared difference

$$\bar{d}_i^2 = \frac{w\, w_i \left(\mu_i - y\right)^2}{(w + w_i)} = \frac{\left(\mu_i - y\right)^2}{\sigma_i^2 + \sigma^2} \tag{5.61}$$

and arrive at

$$a_{i0} = \frac{1}{2\pi\sigma_i\sigma_\theta}\ \exp\left(-\frac{1}{2}\bar{d}_i^2\right)\ \ \exp\left(-\frac{1}{2}\frac{(\theta - \bar{y})^2}{\bar{\sigma}^2}\right) \tag{5.62}$$

Therefore we obtain for the integral of the first two terms

$$\int_\theta a_i \mathrm{d}\theta \quad = \quad P(\mu = \mu_1)\ \mathrm{P_{in}}\ \frac{1}{2\pi\sigma_i\sigma_\theta}\ \exp\left(-\frac{1}{2}\bar{d}_i^2\right)\ \ \int_\theta \exp\left(-\frac{1}{2}\frac{(\theta - \bar{y})^2}{\bar{\sigma}^2}\right)\mathrm{d}\theta \tag{5.63}$$

$$= \quad P(\mu = \mu_1)\ \mathrm{P_{in}}\ \frac{1}{2\pi\sigma_i\sigma_\theta}\ \exp\left(-\frac{1}{2}\bar{d}_i^2\right)\ \ \sqrt{2\pi}\bar{\sigma} \tag{5.64}$$

thus finally at

$$\boxed{A_i = \int_\theta a_i \mathrm{d}\theta = P(\mu = \mu_1)\ \mathrm{P_{in}}\ \frac{1}{\sqrt{2\pi}\sqrt{\sigma_i^2 + \sigma^2}}\ \exp\left(-\frac{1}{2}\frac{\left(\mu_i - y\right)^2}{\sigma_i^2 + \sigma^2}\right)} \tag{5.65}$$

The integral of the second two terms b_{i0} actually are

$$\int_\theta b_{i0}\mathrm{d}\theta = \frac{1}{|a-b|}\left|\int_a^b \mathrm{g}(\theta, \mu_i, \sigma_i^2)\mathrm{d}\theta\right| = \frac{1}{|a-b|}\left|\Phi\left(\frac{b-\mu_i}{\sigma_i}\right) - \Phi\left(\frac{a-\mu_i}{\sigma_i}\right)\right| \tag{5.66}$$

Therefore, we have

$$\boxed{B_i = \int_\theta b_i \mathrm{d}\theta = P(\mu = \mu_i)\ \frac{1 - \mathrm{P_{in}}}{|a-b|}\left|\Phi\left(\frac{b-\mu_i}{\sigma_i}\right) - \Phi\left(\frac{a-\mu_i}{\sigma_i}\right)\right|} \tag{5.67}$$

Part II

Technical Notes on Statistics and Estimation Theory

6 Evaluation of Estimation Results – An Example

We give three examples for parameter estimation and evaluation. We provide the specific models and equations for the examples. The discussion includes general hints how to use the evaluation methods in other applications and how to report evaluation results in publications. The note served as an appendix to Chapt. 4 of Förstner/Wrobel (2016), GC 11, Springer.

6.1 Preface

This note is identical to the Example to Chapter 4 of Förstner and Wrobel (2016) at `https://www.ipb.uni-bonn.de/book-pcv/software/PCV-A-4-examples.pdf` and provides positive and negative examples how to evaluate estimation results.

6.2 Summary

We give three examples for parameter estimation and evaluation

1. Gauss–Markov model for linear regression with two unknowns with evaluation.
2. Gauss–Markov model for linear regression for similarity transformation with evaluation.
3. Gauss–Helmert model for linear regression for similarity transformation with evaluation.

We provide the specific models and equations for the examples.

The discussion includes general hints how to use the evaluation methods in other applications and how to report evaluation results in publications.

Partly we refer to problems addressed in the book. The results with concrete numbers derived with the MATLAB-code, are given in example boxes. The background color

green indicates recommended procedures, red indicates pitfalls, other text has light grey background.

For the theory we refer to Chap. 4. References to sections are given as 'PCV-NUMBER', e.g., PCV-4.2, references to equation as 'PCV-(NUMBER)', e.g., PCV-(4.138). For the software we refer to the home page of the book http://www.ipb.uni-bonn.de/book-pcv/.

6.3 GMM for linear regression with two unknowns with evaluation

The scope of this example is to demonstrate the estimation and the evaluation in the linear Gauss-Markov model. The MATLAB-script file is `GMM/DEMOS-GMM/demos_GMM_regression.m` under http://www.ipb.uni-bonn.de/book-pcv/#cod.

6.3.1 The Model and the Estimates

The observations $y_n, n = 1, ..., N$ depend linearly on the time t. The intercept x_1 and the slope x_2 are unknown, see Fig. 6.1.

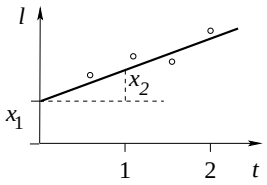


Figure 6.1: Model for Example 1: linear regression

We assume all observations to be uncorrelated and have the same standard deviation $\sigma := \sigma_{y_n}$:

$$\mathbb{E}(\underline{y}_n) = x_1 + x_2 t_n\,, \qquad \mathbb{D}(\underline{y}_n) = \sigma^2\,. \tag{6.1}$$

Collecting the observations, parameters and coefficients in the corresponding vectors and matrices, namely

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \dots \\ y_n \\ \dots \\ y_N \end{bmatrix}\,, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \tag{6.2}$$

and the partitioned design matrix

$$\underset{N\times 2}{A} = \begin{bmatrix} 1 & t_1 \\ \dots & \\ 1 & t_n \\ \dots & \\ 1 & t_N \end{bmatrix} = [\mathbf{1}\ \boldsymbol{t}] := [C, D] \quad \text{with} \quad C = \mathbf{1} \quad \text{and} \quad D = \boldsymbol{t}\,. \tag{6.3}$$

the model reads as

$$\mathbb{E}(\underline{\boldsymbol{y}}) = A\boldsymbol{\theta}\,, \qquad \mathbb{D}(\underline{\boldsymbol{y}}) = \sigma^2 I_N = w^{-1} I_N\,, \tag{6.4}$$

see PCV-(4.34). This is the same model we use for the fitting line in Chap. 10.5 for deriving the uncertainty of the 2D line through given points. The normal equation matrix can be given explicitly

$$N = \begin{bmatrix} Nw & w\sum_{n=1}^{N} x_n \\ w\sum_{n=1}^{N} x_n & w\sum_{n=1}^{N} x_n^2 \end{bmatrix}\,, \tag{6.5}$$

see PCV-(4.40). The estimated residuals are

$$\widehat{\boldsymbol{v}} = (\widehat{\theta}_1 + \widehat{\theta}_2 t_i) - \boldsymbol{y}\,, \tag{6.6}$$

see PCV-(4.41). A numerical example is given in the box on page 123. The estimated sum of the squared residuals and the estimated variance factor are

$$\Omega = w \sum_{n=1}^{N} v_n^2 \quad \text{and} \quad \widehat{\sigma}_0^2 = \frac{\Omega}{N-2}\,, \tag{6.7}$$

see PCV-(4.81) and PCV-(4.80).

The theoretical and the empirical covariance matrices are

$$\Sigma_{\widehat{\theta}\widehat{\theta}} = \sigma_0^2 N^{-1} \quad \text{and} \quad \widehat{\Sigma}_{\widehat{\theta}\widehat{\theta}} = \widehat{\sigma}_0^2 N^{-1}\,. \tag{6.8}$$

The covariance matrix of the residuals is

$$\Sigma_{vv} = \Sigma_{yy} - A\Sigma_{\widehat{\theta}\widehat{\theta}}A^{\mathsf{T}}\,. \tag{6.9}$$

For getting insight into the structure of the result, we reduce the times t_i to their *centroid*

$$\bar{t}_n = t_n - \mu_t \quad \text{with} \quad \mu_t = \frac{\sum_{n=1}^{N} t_n}{N} \tag{6.10}$$

and obtain the design matrix for the centred model

$$A^{(c)} = [1 \;\; \bar{t}_n]_{n=1,\ldots,N}\,. \tag{6.11}$$

Hence the new mode reads as

$$\mathbb{E}(\underline{y}_n) = x_1^{(c)} + x_2 t_i^{(c)}\,, \tag{6.12}$$

where the intercept refers to the abscissa at centroid μ_t. A numerical example is given in the following box. The the covariance matrix of the unknown parameters then is diagonal

$$\Sigma_{\widehat{\theta}\widehat{\theta}} = \frac{\sigma_0^2}{wN\sum_{n=1}^{N}\bar{t}_n^2}\begin{bmatrix} \sum_{n=1}^{N}\bar{t}_n^2 & 0 \\ 0 & N \end{bmatrix} = \frac{\sigma_0^2}{w}\begin{bmatrix} \frac{1}{N} & 0 \\ 0 & \frac{1}{\sum_{n=1}^{N}\bar{t}_n^2} \end{bmatrix} \tag{6.13}$$

Hence the standard deviation of the estimated intercept in the centred model, which is at the centroid, is

$$\sigma_{\widehat{\bar{x}}_1} = \frac{\sigma}{\sqrt{N}}\,, \tag{6.14}$$

which decreases with increasing number N of observations.

Example 6.3.17: ***Linear regression with two parameters (1).*** Figure 6.2 shows the result of an example generated with `demos_GMM_regression.m` and initialization of the random number generator with `init_rand=15`. We will refer to this numerical example in the following.
The true values are given by:

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}, \quad \sigma = 0.5\,, \quad \boldsymbol{t} = \begin{bmatrix} -1 \\ 1 \\ 2 \\ 14 \end{bmatrix}, \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} -0.5 \\ 1.5 \\ 2.5 \\ 14.5 \end{bmatrix}, \quad \tilde{\boldsymbol{e}} = \begin{bmatrix} -0.6271 \\ -0.3932 \\ -0.5604 \\ 3.5861 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 0.1271 \\ 1.8932 \\ 3.0604 \\ 10.9139 \end{bmatrix}. \tag{6.15}$$

The true errors $\tilde{\boldsymbol{e}}$ result from sampling from $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.5$. We also introduce an outlier in order to demonstrate the difficulty to identify outliers. Observation y_4 is changed by the error $\nabla l_4 = -4$; this error is 8 times the standard deviation of the assumed observational noise. The estimated parameters and estimated residuals

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} 1.1956 \\ 0.7008 \end{bmatrix}, \quad \hat{\boldsymbol{v}} = \begin{bmatrix} 0.3678 \\ 0.0032 \\ -0.4633 \\ 0.0922 \end{bmatrix}. \tag{6.16}$$

Observe, this result can be obtained by just assuming the observations have the same standard deviation, though this needs not be known.

If we would have ground truth, i.e., the true values for the parameters, we could report the differences

$$\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}} = \begin{bmatrix} 0.1956 \\ 0.2008 \end{bmatrix}. \tag{6.17}$$

Without knowing anything about the observational process, i.e., the structure of the problem and the level of the observational noise, this difference cannot be evaluated. Moreover, if we – as a reader of such a result – would have a different experimental setup, using the same functional model (here a linear regression with two parameters), e.g., more observations, possibly distributed differently, then we would not be able to predict the performance in our situation. This indicates, that even if we give the differences $\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}$ of the estimates to some ground truth, the reader does not learn something from this difference, if not provided with more information; this will be discussed below.

◇

The standard deviation of the slope is

$$\sigma_{\hat{x}_2} = \frac{\sigma}{\sum_{n=1}^{N} \bar{t}_n^2}. \tag{6.18}$$

With the root mean square distance of the observed times from their centroid

$$\mathrm{RMSE}_t := \frac{1}{N}\sqrt{\sum_{n=1}^{N} \bar{t}_n^2}. \tag{6.19}$$

We hence have the standard deviation of the estimated slope

$$\sigma_{\hat{x}_2} = \frac{\sigma}{\sqrt{N}} \frac{1}{\mathrm{RMSE}_t}. \tag{6.20}$$

A numerical example is given in box on page 126.

6.3.2 The Estimation

The estimation is realized in the MATLABfunction `GaussMarkovModelLinear.m`. It in a first step follows Alg. 1, PCV-p.91. An additional routine `diagnostics_1d.m` performs the sensitivity analysis. Given a set r_U of parameters of interest it determines all diagnostic parameters of interest:

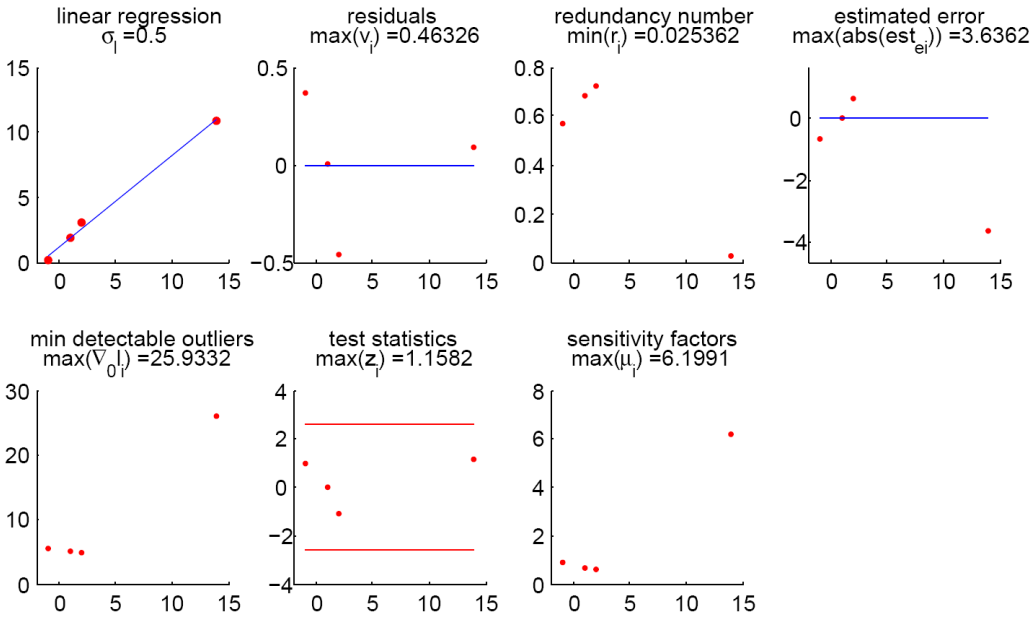


Figure 6.2: Result of regression. From top left to bottom right: (1) original data y_n; (2) estimated residuals $\hat{v}_n$, (3) redundancy numbers r_n; (4) estimated errors ∇y_n; (5) minimal detectable outliers $\nabla_0 y_n$; (6) test statistics z_n; (7) sensitivity factors μ_n. Explanation see text.

- the covariance matrix of the estimated parameters

$$\Sigma_{\hat{\theta}\hat{\theta}} = (A^{\mathsf{T}} W_{yy} A)^{-1} \tag{6.21}$$

  assuming the a priori variance factor is $\sigma_0^2 = 1$. It allows to derive the standard deviations of the estimated parameters $\sigma_{\hat{\theta}_u} = \sqrt{\Sigma_{\hat{\theta}_u \hat{\theta}_u}}$ from the diagonal elements of the covariance matrix. It does not depend on real observations, but only on the mathematical model of the design, i.e., the geometric configuration and the assumed uncertainty of the observations.

- the residuals $\hat{v}_i$ and the estimated varaince factor

$$\hat{v}_n = \boldsymbol{a}_n^{\mathsf{T}} \hat{\boldsymbol{\theta}} + a_n - y_n \quad \text{and} \quad \hat{\sigma}_0^2 = \frac{\hat{\boldsymbol{v}}^{\mathsf{T}} W_{yy} \hat{\boldsymbol{v}}}{N - U}\,. \tag{6.22}$$

- the test statistics z_n (PCV-(4.284))

$$z_n = \frac{-\hat{v}_n}{\sigma_{\hat{v}_n}}\,. \tag{6.23}$$

- the redundancy numbers as diagonal elements of the redundancy matrix (PCV-(4.61))

$$r_n = R_{nn}\,. \tag{6.24}$$

- the minimum size $\nabla_0 y_i$ of detectable outliers (PCV-(4.304), (4.300))

$$\nabla_0 \boldsymbol{y}_i = \delta_0 \frac{\sigma_{y_i}}{\sqrt{r_i}}\,. \tag{6.25}$$

  We use $\delta_0 = 4.13$, see PCV-p.67, Table 3.2.

- the sensitivity factor w.r.t. all 4 parameters (PCV-(4.292))

$$\mu_{x,n} = \sqrt{\frac{1 - r_n}{r_n}}\,; \tag{6.26}$$

- the sensitivity factor w.r.t. the selected set r_U of parameters (in PCV-(4.296) referred to as parameter set $\boldsymbol{\kappa}$)

$$\mu_{x_1,n} = \sqrt{\frac{\overline{u}_n}{r_n}}\,. \tag{6.27}$$

6.3.3 Evaluating the Precision of the Estimates

6.3.3.1 Simulations vs. Theoretical Derivations

We have three methods to derive the theoretical precision of estimates, which in a first step are equivalent:

1. Using the Cramer-Rao bound based on the numerical determination of $\Sigma_{\hat{\theta}\hat{\theta}}$ for well selected cases, see PCV-(4.49). This requires only one simulation and estimation for each configuration.
2. Using the Cramer-Rao bound based on an algebraic derivation. In this case we derive algebraic expressions for the design matrix, the normal equation matrix and its inverse, as in PCV-13.3.6.1 done for the relative orientation of the image pair. This gives direct insight into the dependencies of the standard deviations of the parameters of the configuration.
3. Using sampling techniques as described in PCV-4.6.8.2: For each configuration (choice of the functional and mathematical model) this requires $K > 25$ samples and therefore estimates for obtaining an accuracy of better than 5%.

Depending on the complexity of the problem, we can choose between them.

Example 6.3.18: ***Linear regression with two parameters (2).*** The following information presumes, that some a priori standard deviation σ of the observations is known, i.e., the user of the estimation software knows how accurate the observations are.
All residuals are below σ, hence the result appears fine, though we know there is an outlier in the 4-th observation.

The estimated variance factor is

$$\widehat{\sigma}_0 = \frac{\sum_{n=1}^{N} \widehat{v}_n^2/\sigma_{y_n^2}}{N-U} = 0.8467\,. \tag{6.28}$$

For determining $\widehat{\sigma}_0$ we exploit the assumption, that the observations are mutually uncorrelated. The estimated variance factor $\widehat{\sigma}_0^2$ is not significantly deviating from the a priori value $\sigma_0^2 = 1$. However, since the redundancy $R = N - U = 4 - 2 = 2$ is very low, this value is very uncertain, see the discussion on the estimated variance factor in PCV-4.2.3.

The theoretical covariance matrix and the theoretical standard deviations of the parameters are

$$\Sigma_{\widehat{\theta}\widehat{\theta}} = \begin{bmatrix} +0.0915 & -0.0072 \\ -0.0072 & +0.0018 \end{bmatrix}, \quad \sigma_{\widehat{\theta}_1} = 0.3025\,, \quad \sigma_{\widehat{\theta}_2} = 0.0426\,. \tag{6.29}$$

This is the Cramer-Rao bound, the lower bound for the achievable precision in this experiment, i.e., for this design, the distribution $\boldsymbol{t}$ of the observations, the assumed model and the assumed noise level σ.

The evaluation of the deviations of the estimates $\widehat{\boldsymbol{\theta}} = (1.1956, 0.7008)$ from the ground truth $\tilde{\boldsymbol{\theta}} = (0.5, 1.0)$ now can be related to the theoretical covariance matrix, which depends on both, the design of the experiment and the assumed noise level of the observations. This deviation is significant, since the test statistic (the Mahalanobis distance of $\widehat{\boldsymbol{\theta}}$ from $\tilde{\boldsymbol{\theta}}$)

$$X = (\widehat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}})^{\mathsf{T}} \Sigma_{\widehat{\theta}\widehat{\theta}}^{-1} (\widehat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}) = 53.45 > \chi^2_{2,0.99} = 9.21\,. \tag{6.30}$$

is larger than the tolerance, see the test PCV-(3.32).
Such a comparison is valuable for both the author (having performed the experiment and publishing this in a paper) and the reader (of a conference or journal paper): It tells whether all information of the observations is exploited. For the author this indicates, that there appear not to be any hidden systematic errors left. For the reader this indicates, that the method appears to be adequately designed. This of course has to seen in the context of the size of the experiment, which here is too small.

◇

Before trusting the Cramer-Rao bound and the algebraic derivations, it is useful to perform comparisons between these measures and the result of simulations, in order to get experience for which type of problems simulations appear necessary.

The checks of the implementation as discussed in PCV-4.6.8 are based on the required coherence between the simulations and the other "one-shot" methods using the Cramer-Rao. The evaluation can be based on statistical tests, which allow evaluation as a function also of K. The test only work, if the underlying model is linear enough, i.e., second order effects do not disturb. In order to avoid confusion between different sources for deviations, very small standard deviations (for avoiding second order effects) and large number of iterations (in order to avoid biased estimates) need to be chosen.

A numerical example is given in box on page 127.

Example 6.3.19: ***Linear regression with two parameters (3).*** We want to check the correctness of the implemented software. For this we refer to PCV-4.6.8 and generate a sufficiently large number of samples for the observations for a fixed parameter vector, following the mathematical model of the estimation procedure, hence without outliers. We choose $K = 25$ in order to obtain accurate results for the check. We refer to the result with `init_rand=15`.

The check leads to the following results:

- The mean of the estimated variance factors is $s^2 = 0.6654$. This appears small compared with the expected value $\mathbb{E}(\underline{\hat{\sigma}_0}^2) = 1$. However, the confidence interval $[T_l, T_u]$ for a significance level is $[0.4981, 1.6983]$. Hence we have

$$s^2 = 0.7672 \in [0.5593, 1.5923]\,, \tag{6.31}$$

  and the alternative hypotheses, that the estimated variance factors significantly deviate from 1, is to be rejected.

- The theoretical covariance matrix and the empirical covariance matrix, derived from $K = 25$ estimated parameter vectors $\widehat{\boldsymbol{\theta}}_k$ are

$$\Sigma_{\theta\theta} = \begin{bmatrix} +0.0915 & -0.0072 \\ -0.0072 & +0.0018 \end{bmatrix}, \quad \mathbb{D}(\underline{\widehat{\boldsymbol{\theta}}}) = \begin{bmatrix} 0.0834 & -0.0065 \\ -0.0065 & 0.0021 \end{bmatrix}. \tag{6.32}$$

  The test statistic X^2 (see (4.358)) for checking, whether the estimated covariance matrix significantly deviates from the theoretical covariance matrix, is within the confidence interval:

$$X^2_\Sigma = 0.7528 \in [0.0717, 12.8382]\,. \tag{6.33}$$

- Finally, we check whether the estimated parameters are biased using PCV-(4.360). The mean of the estimated parameters is $\widehat{\boldsymbol{m}}_{\hat{\theta}} = [0.5479, 1.0070]^\mathsf{T}$. The Mahalanobis distance from the true parameter vector $[0.5, 1.0]$ also lies within the confidence region

$$X^2_{\text{bias}} = 1.0756 \in [0.0100, 10.5966]\,. \tag{6.34}$$

Hence, we have no reason to assume the implementation has errors.

◇

6.3.3.2 The Ideal Dependencies

The theoretical precision of the result is representative for many estimation problems. We summarize and interprete these results, and discuss its relevance for other estimation problems.

Recall, the theoretical precision of the estimated parameters is:

$$\sigma_{\hat{\bar{x}}_1} = \frac{\sigma}{\sqrt{N}} \quad \text{and} \quad \sigma_{\hat{x}_2} = \frac{\sigma}{\sqrt{N}} \frac{1}{\text{RMSE}_t} \tag{6.35}$$

- The standard deviations of the estimates σ_{x_u} linearly increase with the standard deviation σ of the observations.
- The standard deviations decreases with the square root of the number N of observations. This strictly only holds for the centroid. The standard deviation of the slope only decreases with $\sqrt{N}$ if the average distance of the observations from the centroid remain unchanged. This holds (approximately) if the density of the observation over time is changed, but the time interval $t_N - t_1$ remains constant. This often is a reasonable model: for example when analysing the absolute or relative orientation of images using well distributed points in the images, then the average spread (RMSE_x) would characterize the distribution of the observed image points independent of the number of image points. Then the standard deviations of the pose parameters will approximately decrease with $1/\sqrt{N}$, where N is he number of points in the image.

The graph of $y = 1\sqrt{N}$ however visually is similar to the graphs of $a = 1/N$ or $y = 1/\log N$. Showing the decay of the standard deviations of the parameters therefore should be accompanied by a graph, showing

$$\sqrt{N}\sigma_{\hat{x}_2} = \frac{\sigma_{\hat{x}_2}}{\mathrm{RMSE}_t} \tag{6.36}$$

This ideally does not depend on the number of observations, if the configuration does not change, only the density of the observations. Deviations easily can be seen.

- The standard deviation of the slope decreases linearly with the width RMSE_t of the data. This is typical for geometric problems, where the observed features "carry" the information: The larger the width of the data, the more precise the solution. The width in structure from motion problems may refer to
  - the coverage of the image area,
  - the viewing angle,
  - the length of a straight line segment, or
  - the area of a planar regions covered by 3D points.

  The special structure of a geometric problem may also lead to other dependencies of the width of the data: As an example: the standard deviation of the rotation angles (ω and φ) of a camera across the viewing direction decrease quadratically with the width d of the image area covered by image features. Here, a plot of $d^2\sigma_\omega$ for varying d should show no dependency on d.

Numerical examples are given in the box on page 129.

6.3.3.3 Causes for Deviations from the Ideal Dependencies

Often these dependencies are derived by simulations to demonstrate the "robustness" of the solution (actually the theoretical precision): showing the uncertainty of the estimated parameters as a function of the noise added to the observations. This is derived by repeating the estimation K times, and reporting the RMSE of the parameters as a function of σ. If the number K of samples is large enough the linear dependency should be visible in the graph.

Deviations from the linearity may either be have different causes, e.g., :

- a too low number K of samples. The relative precision of the estimated standard deviation is appr. $\sqrt{1/K}$. For achieving a 5% accuracy at least $K = 25$ samples need to be taken.
- the influence of the linearization of a non-linear model, see the discussion in Sect. 2.7.6.
- a lack of convergence of an iterative estimation scheme. This may even occur for a linear problem, if no direct solution, e.g., by Gaussian elimination is used to solve the normal equations, but e.g., a conjugate gradient method.

Example 6.3.20: ***Linear regression with two parameters (4).*** We want to demonstrate dependencies of the noise level σ using simulated data, and discuss how to visualize such results.

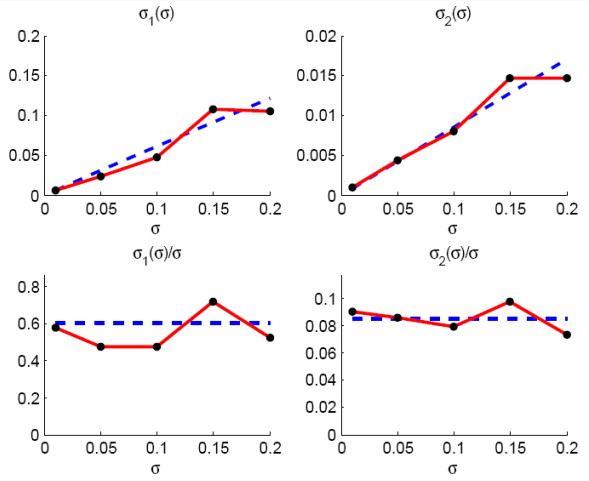
We repeat the simulations used for checking the correctness of the implementation for the following noise levels:

$$\sigma \in [0.01, 0.05, 0.1, 0.15, 0.20]\,, \tag{6.37}$$

and visualize the corresponding standard deviations from the empirically determined covariance matrices (6.32), right. There, for $\sigma = 0.5$ we would obtain $\widehat{\sigma}_{\hat{x}_1} = 0.2889$ and $\widehat{\sigma}_{\hat{x}_2} = 0.0453$. The red lines in the top row of the figure shows the standard deviations $\widehat{\sigma}_{\hat{x}_i}(\sigma), i = 1, 2$ for the two parameters. They approximately increase linearly with σ; the theoretical increase is shown as dashed blue line. The discrepancies are due to the number $K = 25$ of samples used for the simulation, which causes a relative error of 5%. This blue dashed curve usually is not known, unless for a single choice of σ the theoretical covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$ is determined, which for equally weighted observations is

$$\Sigma_{\hat{\theta}\hat{\theta}} = \sigma^2 (A^\mathsf{T} A)^{-1}\,, \tag{6.38}$$

see (PCV-(4.49)). A linear dependency easily can be mistaken for an affine dependency, where there is an offset at $\sigma = 0$, if the simulations do not start with a very small sigma.



Standard deviation of the the parameters as a function of the noise level. The empirical dependency is given in red, the theoretical dependency is given in dashed blue. **Top left:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_1}(\sigma)$. The dependencies should linearly increase. **Top right:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_1}(\sigma)$. **Bottom left:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_2}(\sigma)/\sigma$. **Bottom right:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_2}(\sigma)/\sigma$. The dependencies here should be a constant

If not a very small noise level σ for the observations is included in the simulations, it is recommended to visualize the ratio

$$r_1(\sigma) = \frac{\widehat{\sigma}_{\hat{x}_1}(\sigma)}{\sigma}\,, \tag{6.39}$$

which should be a constant. This easily can be checked visually; see the bottom row.

◇

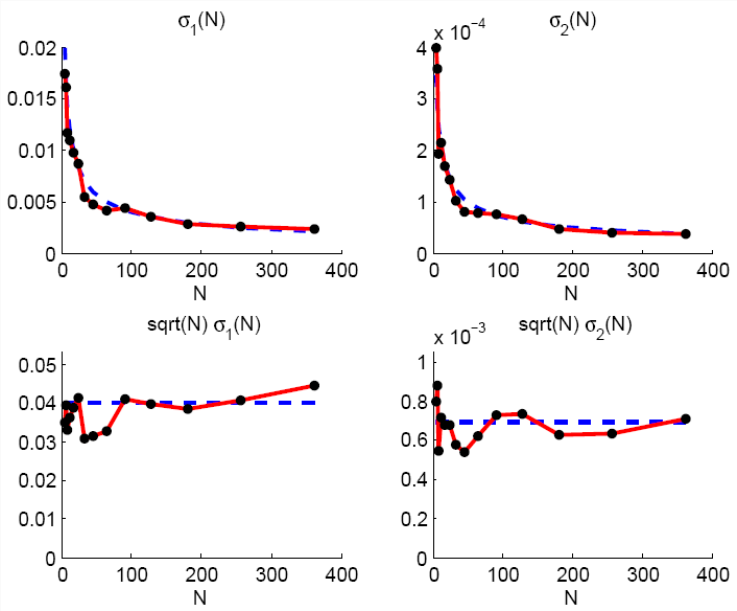
- a Levenberg-Marquardt solution is used in the presence of a singular normal equation system. Hence, the geometry of the problem represents a degenerate configuration. Then the unknown parameters are not estimable, and the regularizer enforces a solution close to the one yielding a minimum norm for the covariance matrix of the unknown parameters. Then the estimates depend on the approximate values. This might lead to a deviation of the linear relationship between the RMSE for the unknown parameters and the assumed noise standard deviation.

Example 6.3.21: ***Linear regression with two parameters (5).*** Similarly as for the noise standard deviation, we want to demonstrate dependencies on the density of the observations using simulated data, and discuss how to visualize such results. We therefore assume the observations to be regularly spaced in a fixed interval, and vary N. We assume this interval to be 100, and the noise standard deviation to be 0.2.

We assume the observations are taken in a fixed interval, and vary N. For the sequence

$$N \in [4, 6, 8, 11, 16, 23, 32, 45, 64, 91, 128, 181, 256, 362] \tag{6.40}$$

the red line in the top row in Fig. 6.3.3.3 shows the estimated $\widehat{\sigma}_{\hat{\theta}_i}, i = 1, 2$. The standard deviations are decaying, as expected.



Standard deviation of the the parameters as a function of observational density The empirical dependency is given in red, the theoretical dependency is given in dashed blue. **Top left:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_1}(N)$. **Top right:** Standard deviation $\widehat{\sigma}_{\hat{\theta}_2}(N)$. The theoretical dependencies follow approximately $1/\sqrt{N}$. **Bottom left:** Standard deviation $\sqrt{N}$
$est\sigma_{\hat{\theta}_1}(N)$. **Bottom right:** Standard deviation $\sqrt{N}\ \ \widehat{\sigma}_{\hat{\theta}_2}(N)$. The dependencies here should be a constant

In order to confirm the dependency on N, namely a decay with $1/\sqrt{N}$, we show

$$r_2(N) = \sqrt{N}\ \ \widehat{\sigma}_{\hat{x}_i}(N)\,, \tag{6.41}$$

in the bottom row. It should be a constant.
Often the level of the theoretical precision is not known or difficult to obtain, e.g., since the software does not provide the standard deviations of the estimates. In this case the blue dashed curves in top row could be replaced by a best fitting function $\sigma_{\hat{\theta}_i} = a_i/\sqrt{N}$ in order to visually prove the type of dependency, and avoid the normalize plots in the bottom row.

◇

- inconsistencies between the simulated data and the used model. Hence the simulation checks both: the program for generating artificial data and the estimation routine.
- suboptimal implementation of the simulation or the estimation.

6.3.4 Testing and the Sensitivity of the Estimation

The evaluation of the result can be based on useful measures for identifying outliers and weaknesses in the geometric configuration, see the box on page 133 and Fig. 6.2, p. 124.

6.3.4.1 Detectability of Outliers and Testing

The largest residual is $\hat{\boldsymbol{v}}_3 = -0.463$, much larger in magnitude than $\hat{v}_4 = 0.092$, though we know that there is an outlier in y_4.

To obtain insight into the geometry of the observational design we investigate the redundancy matrix,[1] which shows how deviations in the observation influence the residuals.

Relating the observations to the centroid the redundancy matrix is

$$R = I - A(A^\mathsf{T}A)^{-1}A = I - \frac{\mathbf{1}\mathbf{1}^\mathsf{T}}{\mathbf{1}^\mathsf{T}\mathbf{1}} - \frac{\bar{\boldsymbol{t}}\bar{\boldsymbol{t}}^\mathsf{T}}{\bar{\boldsymbol{t}}^\mathsf{T}\bar{\boldsymbol{t}}}\,. \tag{6.42}$$

or with the mean position RMSE_t

$$R = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\mathsf{T} + \frac{1}{N}\frac{\bar{\boldsymbol{t}}\bar{\boldsymbol{t}}^\mathsf{T}}{\mathrm{RMSE}_t^2}\,. \tag{6.43}$$

In our case we obtain

$$\nabla\hat{\boldsymbol{v}} = -R\nabla\boldsymbol{y} \quad \text{with} \quad R = \begin{bmatrix} 0.5688 & -0.3587 & -0.3225 & 0.1123 \\ -0.3587 & 0.6848 & -0.2935 & -0.0326 \\ -0.3225 & -0.2935 & 0.7210 & -0.1051 \\ 0.1123 & -0.0326 & \mathit{-0.1051} & 0.0254 \end{bmatrix}. \tag{6.44}$$

Hence, even if the outlier in observation y_4 would have a much larger size, the effect onto the residual of observation y_3 would be larger than the effect onto $\hat{v}_4$. It can be seen from the elements of the redundancy matrix: Since the off-diagonal term $r_{34} = -0.1051$ is approximately 4-times larger than the redundancy number, namely $r_4 = 0.0254 |r_{43}| > r_4$, residual $\hat{v}_3$ is more influence by an outlier in y_4, than the corresponding residual $\hat{v}_4$.

Using the relative distances

$$d_n = \frac{\bar{t}_n}{\mathrm{RMSE}_t} \quad \text{with} \quad |d_n| \le \sqrt{N-1}\,. \tag{6.45}$$

the redundancy numbers r_n thus are

$$r_n = 1 - \frac{1 + d_n^2}{N} = 1 - u_n \in [0, 1]\,. \tag{6.46}$$

They obviously sum to the redundancy $R = N - U = N - 2$.

The redundancy numbers show two extreme distribution.

- If the observations are equally spaced, e.g., $t_n = t_0 + n\Delta t$ then the redundancy of the middle observation (assuming N is odd) $r_{(N+1)/2} = (N-1)/N$, whereas the redundancy number of the first (or the last) observation is $r_1 = (N-1)/N\ (N-2)/(N+1) < r_{(N+1)/2}$. Hence if the number of observations is larger than 40, all redundancy numbers are above 0.9. This simplifies the analysis, and allows to work with approximations. The first approximation for the redundancy numbers

$$r_n \approx \overline{r_n} = \frac{R}{N} \tag{6.47}$$

just assumes the redundancy numbers do not vary too much; this approximation is assumed in the variance analysis using the triangulation, see PCV-15.4.1.3, Eq. (15.132). The second approximation is $r_n \approx 1$ assuming $U \ll N$. Then also $R \approx I$, which simplifies the analysis of the residuals, as they are assumed to be uncorrelated.

[1] The redundancy matrix only is symmetric if all observations have the same weight.

- If $N-1$ of the observations are clustered and one observation is far off, we obtain the maximum redundancy number $r_n \approx 11/(N-1)$ for the observations in the cluster and $r_n \approx 0$ for the observation far off, since it is necessary for determining the slope of the line. Hence, no approximation of the redundancy numbers can be derived, and they need to be used for a reliable analysis.

Testing the residuals is mandatory if outliers are to be expected. The standardized residuals

$$z_n = \frac{-v_n}{\sigma_{v_n}} = \frac{-v_n}{\sigma_{y_n}\sqrt{r_n}}\,, \tag{6.48}$$

via the redundancy number r_n take the geometry into account. They are more sensitive if the redundancy number is small, i.e., at the borders of the observations. A less sensitive test statistic for outlier detection is the normalized residual

$$z_n^* = \frac{-v_n}{\sigma_{y_n}}\,. \tag{6.49}$$

The detectability of outliers can be characterized by the minimum size of an outlier which can be detected reliably by a statistical test. Following PCV-(4.285) and PCV-(4.289) we have this minimum size of a detectable outlier for the two tests

$$\nabla_0 y_n = \delta_0 \frac{\sigma_{y_n}}{\sqrt{r_n}} \quad \text{and} \quad \nabla_0^* y_n = \delta_0 \frac{\sigma_{y_n}}{r_n} = \frac{1}{\sqrt{r_n}} \nabla_0 y_n\,. \tag{6.50}$$

They differ by a factor $1/\sqrt{r_n} \geq 1$. Hence is r_n is small, say below 0.1, we not only see just 10% of the causing outlier in the residuals, but – in our example – instead of $\nabla_0 y_n \approx 12\sigma_n$ for the statistical test we can only find outliers larger than $\nabla_0^* y_n \approx 30\sigma_l$.

6.3.4.2 The Theoretical Sensitivity w.r.t. all Parameters

We now analyse the sensitivity of the result w.r.t. to possible outliers.

The sensitivity factor w.r.t. all parameters is

$$\mu_{\theta,n} = \sqrt{\frac{1-r_n}{r_n}} = \sqrt{\frac{1+d_n^2}{N-(1+d_n^2)}}\,. \tag{6.51}$$

Leverage Points: The effect of observations, besides the number N of observations, essentially depends on the relative distance of the observation to the centroid. Fig. 6.3 shows this dependency for the case $N = 10$. Obviously, observations with small redundancy number have a large influence onto the estimated parameters. Such points are called *leverage points, see PCV p. 127.*

Example 6.3.22: ***Linear regression with two parameters (6).*** We collect the decisive numbers w.r.t. outlier detection. We have the following indicators (see Fig. (6.2), p. 124)

n	v_n	r_n	$\widehat{\nabla y_n}$	z_n	z_n^*	$\nabla_0 y_n$	$\nabla_0^* y_n$
1	0.3678	0.5688	-0.6466	0.9754	0.7356	2.6518	3.5159
2	0.0032	0.6848	-0.0047	0.0078	0.0064	2.4169	2.9206
3	**-0.4633**	0.7210	0.6425	-1.0911	**-0.9265**	2.3554	2.7739
4	0.0922	**0.0254**	**-3.6362**	**1.1582**	0.1844	**12.5584**	**78.8571**

Evaluation of outlier detection. We assumed $\delta_0 = 4$, see PCV, Table 3.2, p. 67

Observe the effect of using a suboptimal, non-sufficient test statistic z_n^*: (1) the test statistic z_n^* (in this example) points towards a wrong observation, (2) outliers must be large by a factor of at least 79 of their standard deviation to be detectable.
Reporting these numbers (except z_n^* and $\nabla_0^* y_n$) for visual inspection of the result may be appropriate for problems with a not too large number of observations. A summarizing report however is useful, where the extreme values are collected together with an indicator whether they are acceptable. These extreme numbers are part of a self-diagnosis of the estimation procedure.

The minimum redundancy number belongs to the 4-th observation:

$$r_4 = 0.025\,. \tag{6.52}$$

It indicates, that if the observation is changed by some amount, the effect onto the corresponding residual is only approximately 2.5% of that amount. The other 97.5% of this amount influence the parameters, as we will see, when analysing the sensitivity of the estimates.

The estimated size of a possible outlier in this observation is

$$\widehat{\nabla y_4} = \frac{-\widehat{v}_4}{r_4} = 3.64\,, \tag{6.53}$$

which is in the right order of magnitude.

A statistical test, does not indicate an outlier: the maximum test statistic occurs at the 4th observation:

$$z_4 = -\frac{\widehat{v}_4}{\sigma_{\widehat{v}_4}} = 1.16 \leq 2.58\,. \tag{6.54}$$

It correctly points towards the erroneous observation, though it is not significant; for a significance level of $S = 99\%$ the two-sided test has a non-rejection region $[-2.58, +2.58]$.

The largest size $\nabla_0 y_n$ of a detectable outlier, when using a statistical test with z_n, is in observation y_4, namely

$$\nabla_0 y_4 = 12.56\,. \tag{6.55}$$

An outlier in this observation needs to be larger than 25 times (!) the standard deviation of $\sigma = 0.5$ to be detectable with a minimum probability of 80%. This three times larger, than the outlier of size $\nabla y_4 = -4$ we introduced.

◇

6.3.4.3 The Theoretical Sensitivity w.r.t. Centroid

We now investigate the sensitivity of then result for the case, that we are only interested in one of the two parameters. We start with the sensitivity w.r.t. the centroid, i.e., the value $f(\mu_t)$ with $f(t) = x_1 + x_2 t$ and $\mu_t = \sum_n t_n/N$.

The question is: How much influence does a non detectable outlier in one of the observations have onto the centroid. Hence the slope of the line is of no interest and treated as a nuisance parameter. This is like we would be only interested in the position of an object in 3D space, and not interested in his orientation (rotation matrix).

We eliminate the scale following PCV-(4.122) and obtain the part $\overline{C}$ of the reduced

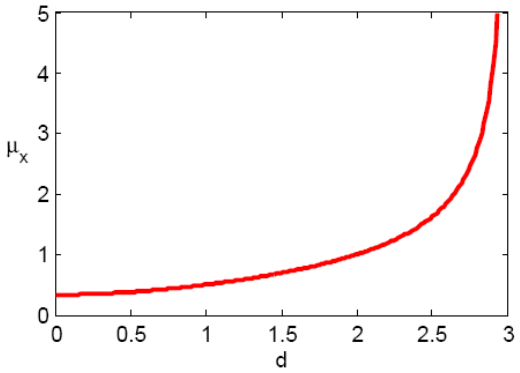


Figure 6.3: Sensitivity factor $\mu_{\theta,n}$ as a function of the relative distance d of the observation from the centroid. For leverage points, i.e., single points lying far apart from the others the relative distance is large. Changes in the corresponding observation have a large influence on the parameters

design matrix:

$$\overline{C} = C = \mathbf{1} \quad \text{and} \quad \Sigma_{x_1 x_1} = \sigma^2 \frac{1}{N} \,. \tag{6.56}$$

We now need the value $\overline{u}_{x_1 n}$, see PCV-(4.128),

$$\overline{u}_{x_1,n} = \frac{1}{N} \,. \tag{6.57}$$

Therefore the sensitivity factor w.r.t. to the centroid is, see PCV-(4.296)

$$\mu_{x_1,n} = \sqrt{\frac{\overline{u}_n}{r_n}} = \sqrt{\frac{1}{N - (1 + d_n^2)}} \leq \mu_{\theta,n} \,. \tag{6.58}$$

Figure 6.4 shows the dependency of the sensitivity factor $\mu_{x_1,n}$ on the relative distance of an observation to the centroid. It is significantly smaller than $\mu_{x,n}$, since parts of the non detectable errors are absorbed by the slope, which is a nuisance parameter.

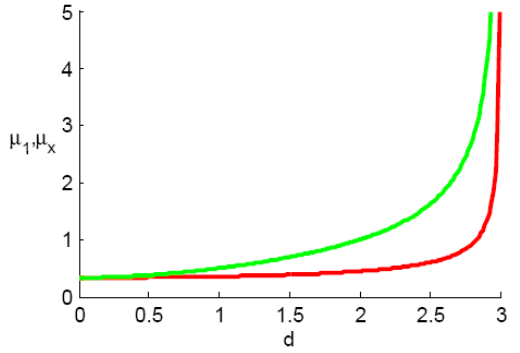


Figure 6.4: Sensitivity w.r.t. the centroid (red) compared to the sensitivity w.r.t. all parameters (green). Moderate leverage points mainly influence the slope, hence have only a limited influence onto the centroid. Only in extreme situations, where the distance of a point is very far off, non-detectable outliers are large enough to still have an influence on the centroid

6.3.4.4 The Theoretical Sensitivity w.r.t. Slope

In a similar manner we can analyse the sensitivity of the estimated slope. This is similar to analysing the sensitivity of the estimated rotation of an object, observed by a motion capture system, taking the 3D coordinates of the centre of gravity as nuisance parameters.

Here we reduce the normal equation system to the slope, and obtain the reduced design matrix

$$\overline{D} = D = \boldsymbol{t} \quad \text{and} \quad \Sigma_{x_2 x_2} = \sigma^2 \frac{1}{N\mathrm{RMSE}_t^2} \,. \tag{6.59}$$

Here we now need $\overline{u}_{x_2 n}$:

$$\overline{u}_{x_2,n} = \frac{\overline{t}_n^2}{N\mathrm{RMSE}_t^2} = \frac{d^2}{N} \,. \tag{6.60}$$

Therefore the sensitivity factor w.r.t. to the slope is

$$\mu_{x_2,n} = \sqrt{\frac{d^2}{N - (1 + \delta_n^2)}} \leq \mu_{\theta,n} \tag{6.61}$$

The dependency of the sensitivity factor $\mu_{x_2,n}$ on the relative distance of an observation to the centroid is shown in Fig. 6.4. Obviously the difference is largest for points close to the centroid, reducing the sensitivity factor to 0: this is plausible, since these observations have no influence on to the slope at all.

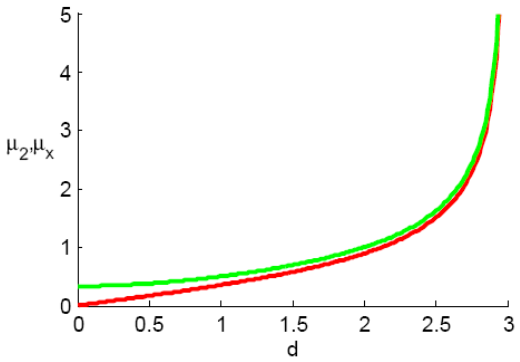


Figure 6.5: Sensitivity w.r.t. the slope (red) compared to the sensitivity w.r.t. all parameters (green). Points close to the centroid have no influence on the slope, as to be expected

6.4 Gauss–Markov Model for Planar Similarity Transformation with Evaluation

This section gives more details on the estimation of a similarity transformation used for generating Fig. 4.11. It at the same time explains the corresponding MATLAB source file `fig_4_11_test_sensitivity_factors_GMM_similarity.m`.

6.4.1 The Mathematical Model

The geometric model is the following

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} ax_i - by_i + c \\ bx_i + ay_i + d \end{bmatrix} \,. \tag{6.62}$$

It holds for the true or expected values.

We assume the coordinates $\boldsymbol{x}_i = [x_i, y_i]^\mathsf{T}$ are given fixed values, the transformed coordinates $\boldsymbol{x}_i' = [[x_i', y_i]^\mathsf{T}$ are observed and the 4 parameters $[a, b, c, d]$ are unknown. We assume the observed coordinates have the same uncertainty, with covariance matrix $\Sigma_{x_i' x_i'} = \sigma^2 I_2$. Figure 6.6 shows

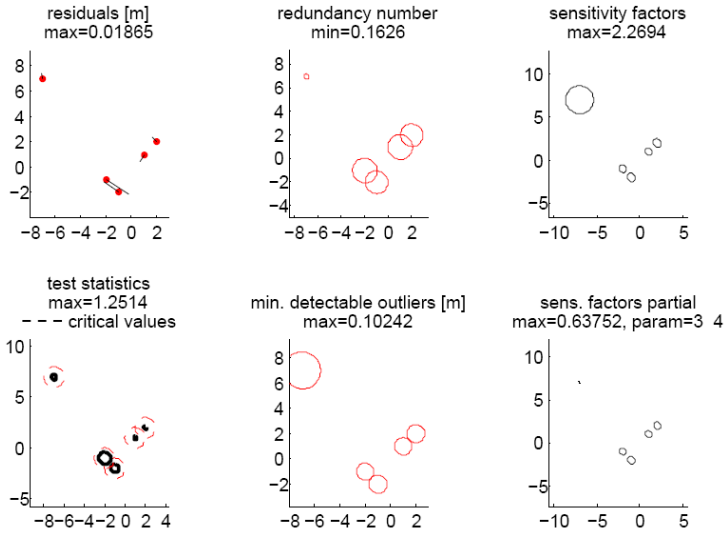


Figure 6.6: Results and diagnostic parameters for a similarity transformation, $\sigma = 0.01$ m.

We collect the $N = 2I$ observations (lines 93/94) and the $U = 4$ unknown parameters (lines 69) in the vectors

$$\boldsymbol{y} := \begin{bmatrix} \boldsymbol{x}'_1 \\ \dots \\ \boldsymbol{x}'_i \\ \dots \\ \boldsymbol{x}'_I \end{bmatrix} \quad \text{and} \quad \boldsymbol{\theta} := \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} . \tag{6.63}$$

The $N \times U$ design matrix is (see lines 95/96) is

$$A = [A_i^\mathsf{T}] = \left[\underbrace{\begin{bmatrix} x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{bmatrix}}_{A_i^\mathsf{T}} \right|_{i=1,\dots,I} . \tag{6.64}$$

The Gauss–Markov model reads

$$\boldsymbol{y} + \boldsymbol{v} = A\boldsymbol{\theta} + \boldsymbol{a} , \qquad \Sigma_{yy} = \sigma^2 I_N , \tag{6.65}$$

with the constant vector $\boldsymbol{a} = \boldsymbol{0}$. Observation wise this is

$$\boldsymbol{y}_i + \boldsymbol{v}_i = A_i^\mathsf{T}\boldsymbol{\theta} , \qquad \Sigma_{y_i y_i} = \sigma^2 I_2 , \tag{6.66}$$

A numerical example is given in the box on 137

Example 6.4.23: ***Similarity transformation (1).*** Figure 6.6 shows the result of a similarity transformation using `GaussMarkovModelLinear_groups.m` with initialization of the random numbers with `init_rand`= 15. We used this configuration for generating the images in PCV-Fig.4.11. We will refer to this figure in the following. The true values are given by:

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} 2.0 \\ 0.5 \\ 3.0 \\ -2.0 \end{bmatrix}, \quad \sigma = 0.01\,, \quad [(x_i, y_i)] = \begin{bmatrix} -7.0 & 7.0 \\ 1.0 & 1.0 \\ 2.0 & 2.0 \\ -1.0 & -2.0 \\ -2.0 & -1.0 \end{bmatrix}, \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} -14.5 & 8.5 \\ 4.5 & 0.5 \\ 6.0 & 3.0 \\ 2.0 & -6.5 \\ -0.5 & -5.0 \end{bmatrix}. \tag{6.67}$$

The true errors $\tilde{\boldsymbol{e}}$ are generated as sample from $\mathcal{N}(0, \sigma^2)$.

$$\tilde{\boldsymbol{e}} = \begin{bmatrix} 0.0159 & 0.0003 \\ 0.0033 & 0.0121 \\ 0.0026 & 0.0038 \\ 0.0108 & 0.0072 \\ -0.0082 & 0.0204 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} -14.4841 & 8.5003 \\ 4.5033 & 0.5121 \\ 6.0026 & 3.0038 \\ 2.0108 & -6.4928 \\ -0.5082 & -4.9796 \end{bmatrix}. \tag{6.68}$$

The estimated parameters and the residuals, shown in Fig. 6.6 upper left, are

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} 1.9986 \\ 0.4996 \\ 3.0023 \\ -1.9899 \end{bmatrix}, \quad \hat{\boldsymbol{v}} = \begin{bmatrix} -0.0008 & 0.0030 \\ -0.0019 & -0.0038 \\ -0.0022 & 0.0027 \\ -0.0079 & 0.0061 \\ 0.0129 & -0.0080 \end{bmatrix}. \tag{6.69}$$

◇

6.4.2 The Estimation

The estimation is realized in the MATLAB function `GaussMarkovModelLinear_groups.m`. It in a first step follows Alg. 1, PCV-p.91. An additional routine `diagnostics_GMM_multi_d.m` performs the sensitivity analysis. Given a set r_U of parameters of interest it determines all diagnostic parameters of interest:

- the covariance matrix of the estimated parameters

$$\Sigma_{\hat{\theta}\hat{\theta}} = (A^{\mathsf{T}} W_{yy} A)^{-1} \tag{6.70}$$

  assuming the a priori variance factor is $\sigma_0^2 = 1$.

- the residuals $\hat{\boldsymbol{v}}_i$ and the estimated variance factor

$$\hat{\boldsymbol{v}}_i = A_i^{\mathsf{T}} \hat{\boldsymbol{\theta}} + \boldsymbol{a}_i - \boldsymbol{y}_i \quad \text{and} \quad \hat{\sigma}_0^2 = \frac{\hat{\boldsymbol{v}}^{\mathsf{T}} W_{yy} \hat{\boldsymbol{v}}}{N - U} \tag{6.71}$$

- the test statistics X_i (PCV-(4.302))

$$X_i = \hat{\boldsymbol{v}}_i^{\mathsf{T}} \Sigma_{\hat{v}_i \hat{v}_i}^{-1} \hat{\boldsymbol{v}}_i\,. \tag{6.72}$$

- the diagonal $d \times d$ block R_{ii} of the the redundancy matrix R (PCV-(4.299))

$$R_{ii} = \Sigma_{\hat{v}_i \hat{v}_i} W_{y_i y_i}\,. \tag{6.73}$$

- the minimum size $\nabla_0 \boldsymbol{y}_i$ of detectable outliers (PCV-(4.304), (4.300))

$$\nabla_0 \boldsymbol{y}_i = \delta_0 \sqrt{\lambda_{\max}(R_{ii}^{-1} \Sigma_{y_i y_i})}\,. \tag{6.74}$$

  We use $\delta_0 = 4.13$, independent on the dimension. This is a useful choice if the group size of the observations is not too large. As can be seen in Table 3.3, PCV-p.68, in our case of $d = 2$ this corresponds to applying a test with significance number $\alpha = 0.3\%$ and requiring a minimum power of $\beta_0 = 80\%$ for finding an outlier.

- the sensitivity factor w.r.t. all 4 parameters

$$\mu_{x,n} = \sqrt{\lambda_{\max}(R_{ii}^{-1} - I_d)}\,; \tag{6.75}$$

From PCV-(4.310) we have $\Sigma_{\hat{y}_i\hat{y}_i}\Sigma_{\hat{v}_i\hat{v}_i}^{-1} = (\Sigma_{y_iy_i} - \Sigma_{\hat{v}_i\hat{v}_i})\Sigma_{\hat{v}_i\hat{v}_i}^{-1} = \Sigma_{y_iy_i}\Sigma_{\hat{v}_i\hat{v}_i}^{-1} - I_2 = R_{ii}^{-1} - I_2$.

- the sensitivity factor w.r.t. the selected set r_U of parameters (in PCV-(4.315) referred to as parameter set $\boldsymbol{\kappa}$)

$$\mu_{x_1,n} = \sqrt{\lambda_{\max}(\overline{U}_{r_U,ii}\, R_{ii}^{-1})}\,. \tag{6.76}$$

This holds since $\overline{U}_{r_U,ii} = \overline{C}_i^{\mathsf{T}}\Sigma_{r_Ur_U}\overline{C}_i W_{y_iy_i}$ (see PCV-(4.125))

The main results for the example are collected in the box on 140. It explicitly addresses tools for self-diagnosis:

- The covariance matrix or the standard deviations of the estimated parameters tell the sensitivity of the result w.r.t. randm errors in the observations.
- The estimated variance factor indicates the overall consistency of the model with the data. Here it can be determined from

$$\hat{\sigma}_0^2 = \frac{\sum_{i=1}^N \hat{\boldsymbol{v}}_i^{\mathsf{T}} W_{\hat{v}_i\hat{v}_i}\hat{\boldsymbol{v}}_i}{N-U} = \frac{\sum_{i=1}^N |\hat{\boldsymbol{v}}_i|^2/\sigma^2}{2I-4} \quad \text{with} \quad \underline{\hat{\sigma}_0}^2 \sim F_{N-U,\infty}\,. \tag{6.77}$$

The first expression for determining the estimated variance factor assumes the observational groups $\boldsymbol{y}_i$ to be uncorrelated, but may have individual and full covariance matrices $\Sigma_{y_iy_i} = W_{y_iy_i}^{-1}$. The second expression exploits the assumption that all points have the same isotropic uncertainty $\Sigma_{y_iy_i} = W_{y_iy_i}^{-1} = \sigma^2 I_2$. cFor a discussion on the evaluation of the estimated variance factor see PCV-4.2.3.

- The *maximal residual* $\max(\boldsymbol{v}_n)$ (or $\max(|\boldsymbol{v}_i|)$) should always be reported, though, if the design is not homogeneous it does not tell whether there are no outliers. Observe, if the observations have the same standard deviation σ and are uncorrelated, the RMSE_v of the residuals (the dimension of the observational groups is d)

$$\mathrm{RMSE}_v = \sqrt{\frac{1}{N}\sum_{n=1}^N v_n^2} = \sqrt{\frac{1}{dI}\sum_{i=1}^N |\boldsymbol{v}_i|^2} \tag{6.78}$$

is related to the variance factor by

$$\hat{\sigma}_0^2 = \frac{N}{N-U}\frac{1}{\sigma^2}\,\mathrm{RMSE}_v^2\,, \tag{6.79}$$

a result which allows us to *statistically test the root mean square error.* If the observations have different weight or are correlated, the RMSEdoes not follow a χ^2-distribution; it does not lead to a sufficient test statistic, since the prior knowledge about stochastical model is not used.

- The *maximal redundancy number* $\min(r_n)$ should be above 0.1.
- The *maximum test statistic* $\max(z_n)$ should always be reported in order to be sure that the statistical test dose not suggest an outlier to be present, if it remains in the non-rejection region which can be derived from the χ^2-distribution.
- The *maximum just detectable outlier* $\max(\nabla_0 y_n)$ indicates in a application oriented way whether the geometry allows to identify outliers. Observe, the value 0.102 is more than 10 times the standard deviation. This measure only is relevant if the goal of the estimation is to find outliers.

- The *maximum sensitivity factor* $\max(\mu_{x,n})$ should always be reported. It should be lass than5 or 10, since then non-detectable outliers have an influence of less than 20 or 40 times the standard deviation of the resultant estimates.
- The *maximum sensitivity factor* $\max\left(\mu_{x_{r_U},n}\right)$ is very useful if the goal of the estimation is to estimate the parameters in r_U, and the user wants to have a guarantee that non-detectable outliers do not perturb the result. Observe, this sensitivity factor may be small, even if quite large outliers may stay undetected.

Such a summarizing self-diagnosis is useful for a quick evaluation of the quality of the result of the estimation. Visualizing the individual numbers, as in Fig. 6.6, of course needs to be adapted to the individual estimation problem. If the estimation process is one module within a chain of modules, the characterizing numbers may be used by the subsequent module.

Example 6.4.24: ***Similarity transformation (2).*** The quality of the result of the estimation needs to be documented. The covariance matrix of the estimates is

$$\Sigma_{\hat{\theta}\hat{\theta}} = 10^{-4} \begin{bmatrix} 0.0102 & 0 & 0.0142 & -0.0142 \\ 0 & 0.0102 & 0.0142 & 0.0142 \\ 0.0142 & 0.0142 & 0.2398 & 0 \\ -0.0142 & 0.0142 & 0 & 0.2398 \end{bmatrix} \tag{6.80}$$

Observe, the parameters $\widehat{x}_1 = \widehat{a}$ and $\widehat{x}_2 = \widehat{b}$ representing scale and rotation are uncorrelated, also the two translation parameters $\widehat{x}_3 = \widehat{c}$ and $\widehat{x}_4 = \widehat{d}$ are uncorrelated. This is caused by the assumption that the observed points are uncorrelated and have the same standard deviation, thus the uncertainty of the point group is isotropic.

The standard deviations of the parameters are

$$\sigma_{\hat{x}_1} = \sigma_{\hat{x}_2} = 0.0010\,, \quad \sigma_{\hat{x}_3} = \sigma_{\hat{x}_4} = 0.0049\,. \tag{6.81}$$

The estimated variance factor is

$$\widehat{\sigma}_0^2 = 0.7863\,. \tag{6.82}$$

The quality of the observations can be characterized by the following statements, which are taken from the MATLAB output:

```
Maximal residual ......................  = 0.01520 at observation 5
Minimal redundancy number .............  = 0.16260 at observation 1
Maximal test statistic ................  = 1.25145 at observation 5
Maximum of minimal detectable outlier ..  = 0.10242 at observation 1
Maximal sensitivity factor ............  = 2.26936 at observation 1
Maximal sensitivity factor translation .  = 0.63752 at observation 4
```

Here we have assumed the user is also interested in the sensitivity of the estimated translation parameters x_3 and x_4 only, see PCV-Fig.4.11 right.

The quality measures indicate, that there is no reason to assume the model not to be consistent with the data and that the geoemtric configuration is acceptable.

◇

6.5 Gauss–Helmert Model for Planar Similarity Transformation

This section has two goals:

1. Demonstrate the sensitivity analysis for the Gauss–Helmert model.
2. Discuss the conditions for the equivalence of the Gauss–Helmert model and the Gauss–Markov model.

6.5.1 Sensitivity Analysis for the Gauss–Helmert Model

The sensitivity analysis aims at investigating the ability to find outliers in the observations and to determine the effect of non-detectable errors on the estimated parameters.

Let the group of observations related to the i-th constraint be collected in the vector $\widehat{\boldsymbol{v}}_i$. We assume that constraints do not share observations, see PCV-4.8.2.5. The direct approach would require the inversion of the covariance matrix $\Sigma_{\hat{v}_i\hat{v}_i}$ of the corresponding group of estimated residuals. Let the size of this group be N_i.

It can be derived from PCV-(4.456)

$$\Sigma_{\hat{v}\hat{v}} = \Sigma_{yy} B W_{gg} (Z^{\mathsf{T}} \Sigma_{yy} Z - X \Sigma_{\hat{\theta}\hat{\theta}} X^{\mathsf{T}}) W_{gg} B^{\mathsf{T}} \Sigma_{yy}\,. \tag{6.83}$$

Assuming $\Sigma_{yy} = \mathrm{Diag}(\{\Sigma_{y_i y_i}\})$ the covariance matrix related to the i-th group is

$$\underbrace{\Sigma_{\hat{v}_i\hat{v}_i}}_{N_i \times N_i} = \underbrace{\Sigma_{y_i y_i} B_i^{\mathsf{T}}}_{N_i \times G} \underbrace{Z}_{G \times G} \underbrace{B_i^{\mathsf{T}} \Sigma_{y_i y_i}}_{G \times N_i} \quad \text{with} \quad Z = W_{gg} (Z^{\mathsf{T}} \Sigma_{yy} Z - X \Sigma_{\hat{\theta}\hat{\theta}} X^{\mathsf{T}}) W_{gg}\,. \tag{6.84}$$

Since generally $G < N_i$ the rank of this $N_i \times N_i$ is only G, thus it cannot be inverted.

However, testing the group $\boldsymbol{y}_i$ related to the i-th set of constraints is equivalent to testing the residual $\boldsymbol{c}_{g,i}$ of that constraint. This equivalent to use the residual

$$\boldsymbol{v}_g = \mathsf{Z}^\mathsf{T}\boldsymbol{v} \tag{6.85}$$

and use the linearized Gauss–Markov model (see PCV-457)

$$\boldsymbol{g}(\boldsymbol{y}, \hat{\boldsymbol{\theta}}^a) + \hat{\boldsymbol{v}}_g = -\mathsf{X}\widehat{\Delta\boldsymbol{\theta}} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{v}}_g) = \mathsf{Z}^\mathsf{T}\boldsymbol{\Sigma}_{yy}\mathsf{Z}\,, \tag{6.86}$$

for testing w.r.t. outliers in the values $\boldsymbol{c}_{g,i}$. This type of diagnosis is realized in the MATLAB function `diagnostics_GHM_constraints_multi_d.m`.

6.5.2 Estimation a Similarity Transformation using the Gauss–Helmert Model

If the observations can be expressed as functions of the unknown parameters, the Gauss–Markov model is the most appropriate model for estimation. If the similarity transformation has to be estimated from point pairs, which both are observed, we obtain constraints between the observations (x_i, y_i, x'_i, y'_i) and the unknown parameters (a, b, c, d):

$$\boldsymbol{g}_i := \begin{bmatrix} \tilde{x}'_i \\ \tilde{y}'_i \end{bmatrix} - \left(\begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{6.87}$$

This constraint not easily can be transformed into a Gauss–Markov model. Therefore, here the Gauss–Helmert model is the appropriate choice for estimation.

The model is linear in the unknown parameters and, starting from approximate values zero, can be written as

$$\boldsymbol{g}(\hat{\boldsymbol{y}}_i^a, \hat{\boldsymbol{\theta}}^a) + \mathsf{X}_i^\mathsf{T}\widehat{\Delta\boldsymbol{\theta}} + \mathsf{Z}_i\widehat{\Delta\boldsymbol{y}_i} = \boldsymbol{0}\,, \quad \mathbb{D}(\underline{\boldsymbol{y}}_i) = \boldsymbol{\Sigma}_{y_i y_i}\,, \tag{6.88}$$

with

$$\boldsymbol{\theta} := \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \quad \boldsymbol{y}_i = \begin{bmatrix} x_i \\ y_i \\ x'_i \\ y'_i \end{bmatrix}, \tag{6.89}$$

and

$$\boldsymbol{g}_i := \boldsymbol{0}\,, \quad \mathsf{X}_i^\mathsf{T} := \begin{bmatrix} -x_i & y_i & -1 & 0 \\ -y_i & -x_i & 0 & -1 \end{bmatrix}, \quad \mathsf{Z}_i^\mathsf{T} := \begin{bmatrix} -a & +b & 1 & 0 \\ -b & -a & 0 & 1 \end{bmatrix}. \tag{6.90}$$

The covariance matrix of the observations is assumed to be block diagonal, i.e., the observational groups are mutually uncorrelated. However, the covariance matrix of each group may contain arbitrary correlations. What is relevant in our context, the covariance matrix also may be singular, as long as the covariance matrix $\boldsymbol{\Sigma}_{g_i g_i} = \mathsf{Z}_i^\mathsf{T}\boldsymbol{\Sigma}_{y_i y_i}\mathsf{Z}_i$ is regular.

Hence we can simulate the model used in the previous section, where the coordinates (x_i, y_i) are assumed to be fixed, non-stochastic values by using the covariance matrix

$$\mathbb{D}(\boldsymbol{y}_i) = \mathbb{D}\left(\begin{bmatrix} \underline{x}_i \\ \underline{y}_i \\ \underline{x}'_i \\ \underline{y}'_i \end{bmatrix} \right) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{x'_i} & \sigma_{x'_i y'_i} \\ 0 & 0 & \sigma_{x'_i y'_i} & \sigma^2_{y'_i} \end{bmatrix}. \tag{6.91}$$

The estimation process does not need the inverse covariance matrix of the observations, therefore this way of modelling fixed observations does not lead to numerical difficulties.

The MATLAB function `demo_GHM_similarity.m` allows to simulate the result of the Gauss–Markov model with the boolean variable `simulate_GMM_similarity=true`, yielding the same result. Since the model is linear, only one iteration needs to be performed.

7 Gauss–Helmert Model as Optimization Problem

The Gauss–Helmert generalizes the well-known Gauss–Markov model by allowing implicit relations between the observations and the unknown parameters. The classical derivation of the estimation procedure refers to the statistical nature of the Maximum-Likelihood optimization. The note separates the description of the model and the optimization function from the generally iterative numerical optimization procedure, in order to elucidate the non-statistical properties of the intermediate steps before treating point of convergence as final estimate.

Remark: While throughout the notes we use one of the classical statistical notation (observations $\boldsymbol{y}$, and parameters $\boldsymbol{\theta}$) , in this note we adopt one of the notations used in Geodesy and Photogrammetry which better fits to the notation used by Boyd and Vandenberghe (2004), thus we name the observations $\boldsymbol{l}$ and the unknown parameters $\boldsymbol{x}$. ◇

7.1 Preface

This note (2021) describes the estimation within the Gauss–Helmert model as a specific optimization problem, making explicit the numerical character of the numerical process for determining the parameters, omitting the statistical interpretation of the intermediate steps within the optimization procedure. This clarifies (1) the role of the stochastical model at the beginning of statistical parameter estimation task and used for evaluating the uncertainty of the result, and (2) the non-statistical role of the numerical method for achieving the final parameters. It is common to derive the estimator for a parameter vector within a statistical framework, and not distinguish the different aspects of the whole task: (a) the specification of the model, (b) the specification of the optimization function, (c) the numerical process of optimization, and (d) the evaluation of the obtained parameters. This note is intended to separate these steps.

7.2 Motivation

Parameter estimation consists in determining unknown parameters from given observations. Its mathematical model consists of the functional model, relating the mean values of the observations to the unknown parameters, and the stochastical model which describes the uncertainty of observation process. We often categorize functional models according to their algebraic structure. The *Gauss–Markov model* is a functional model, where the mean observations are an *explicit* function $\mathbb{E}(\underline{\boldsymbol{l}}) = \boldsymbol{f}(\boldsymbol{x})$ of the parameters. *Gauss–Markov model*

Here we discuss the mathematical model of an estimation task with a functional model, where the mean observations and the parameters are related by an *implicit* function This is called the *Gauss–Helmert model*. Given are N observations $\boldsymbol{l}$ together with the uncertainty of the observation process $\mathbb{D}(\underline{\boldsymbol{l}})$, implicitly assuming the measuring deviations are normally distributed. The mean values $\mathbb{E}(\underline{\boldsymbol{l}})$ of the observations are functionally related to unknown parameters $\boldsymbol{x}$ by G implicit equations *Gauss–Helmert model*

$$\boldsymbol{g}(\boldsymbol{x}, \mathbb{E}(\underline{\boldsymbol{l}})) = \boldsymbol{0}\,. \tag{7.1}$$

The task is to find optimal estimates $\boldsymbol{x}$ for the unknown parameters.

The derivation, presented here, is based on the following assumptions.

- We consider the cases where the representation of the parameters and observations may be redundant, such as for normalized homogeneous coordinates or rotation matrices. Instead of including constraints, such as a length or an orthogonality constraint, we allow that the estimation refers to a minimal representation of the corrections, close to the approximate values of the parameters or the observations, namely in the tangent space defined by the individual constraints. As a consequence, the observations and parameters may be lists of individual groups of possibly redundantly represented entities, e.g., $\boldsymbol{x} := \{R, \boldsymbol{t}, \lambda)$ for the rotation, the translation, and the scale of a spatial similarity, the corrections, however, are vectors of a locally minimal representation, e.g., $\Delta\boldsymbol{x} = [\Delta\boldsymbol{r}^\mathsf{T}, \Delta\boldsymbol{t}^\mathsf{T}, \Delta\lambda]^\mathsf{T}$, where $\Delta\boldsymbol{r}$ describes a small rotation with three parameters.
- We treat the expectation of the observations $\boldsymbol{y} = \mathbb{E}(\underline{\boldsymbol{l}})$ as unknowns. This is a consequence of the previous point and in contrast to classical setups, where the optimization function has the residuals as unknown. In the linearized model the corrections $\Delta\boldsymbol{y}$ and $\Delta\boldsymbol{x}$ to the expectation of the observations $\mathbb{E}(\underline{\boldsymbol{l}})$ and the parameters $\boldsymbol{x}$ are unknown, which allows us to update them in the original, non-linear model taking their algebraic properties, e.g., length or orthogonality, into account.

7.3 The Gauss–Helmert model for estimating parameters

We now describe the set-up of the estimation procedure with a Gauss–Helmert model as functional model, derive the optimization task, provide a solution for the case where the model is linear, finally provide the solution to the non-linear model using a linearized model within an iterative scheme.

7.3.1 The mathematical model

We start from N given observations, collected in the N-vector $\boldsymbol{l}$. We assume, they are a sample of a normal distribution, specified by the *unknown expectation vector and partially known dispersion matrix*. The *stochastical model* for the observation process therefore is given by *stochastical model*

$$\underline{\boldsymbol{l}} \sim \mathcal{N}(\mathbb{E}(\underline{\boldsymbol{l}}), \mathbb{D}(\underline{\boldsymbol{l}}))\,, \tag{7.2}$$

The dispersion matrix of the observations *variance factor* σ_0^2

$$\mathbb{D}(\underline{\boldsymbol{l}}) = \sigma_0^2\,\Sigma_{ll}\,. \tag{7.3}$$

is specified by an approximate covariance matrix Σ_{ll} which differs from the true covariance matrix by an *unknown variance factor* σ_0^2. The *functional model* of the Gauss–Helmert assumes the U unknown parameters $\boldsymbol{x}$ and the N unknown mean values $\mathbb{E}(\underline{\boldsymbol{l}})$ are constrained by the following G-dimensional implicit function[1][2] *functional model*

$$\underset{G\times 1}{\boldsymbol{g}}\,(\underset{U\times 1}{\boldsymbol{x}}, \underset{N\times 1}{\mathbb{E}(\underline{\boldsymbol{l}})}) = \mathbf{0}\,. \tag{7.4}$$

Observe, that (7.2) can be interpreted as the likelihood function of the unknown parameters $\boldsymbol{x}$ *likelihood function*

$$L(\boldsymbol{x}) := L(\boldsymbol{x}, \boldsymbol{g}) = p(\boldsymbol{l} \mid \boldsymbol{x}, \boldsymbol{g}) = \mathcal{M}(\mathbb{E}(\underline{\boldsymbol{l}} \mid \boldsymbol{x}, \boldsymbol{g}), \mathbb{D}(\underline{\boldsymbol{l}} \mid \boldsymbol{x}, \boldsymbol{g}))\,, \tag{7.5}$$

for given observations $\boldsymbol{l}$ and functions $\boldsymbol{g}$, where the distribution $\mathcal{M}$ is characterized by its first and second moment. In order to be able to determine the U parameters $\boldsymbol{x}$ we need to require there are at least as many constraints as unknowns:

$$G \geq U\,, \tag{7.6}$$

or that the number of redundant constraints, i.e., the redundancy *redundancy*

$$R = G - U \geq 0\,. \tag{7.7}$$

is non-negative. Similarly, in order to have a guarantee that the implicit function (7.4) of $[\boldsymbol{x}^\mathsf{T}, \mathbb{E}(\underline{\boldsymbol{l}}^\mathsf{T})] \in \mathbb{R}^{U+N}$ is not empty, the number G of constraints should not exceed $U + N$, hence

$$N \geq G - U \tag{7.8}$$

Therefore we have the following relation

$$N \geq R \geq 0 \tag{7.9}$$

as a necessary condition for the model setup.

7.3.2 The task

The goal is to find the maximum-likelihood estimates $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{y}}$ for the unknown parameters $\boldsymbol{x}$ and the unknown expectation of the observations, short, the *mean observations* $\boldsymbol{y} = \mathbb{E}(\underline{\boldsymbol{l}})$ such that the weighted sum of the residuals[3] *unknown parameters $\boldsymbol{x}$ and unknown mean observations $\boldsymbol{y}$*

$$\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{y}} - \boldsymbol{l}\,, \tag{7.10}$$

namely

$$\Omega = \boldsymbol{v}^\mathsf{T}\Sigma_{ll}^{-1}\boldsymbol{v}\,, \tag{7.11}$$

becomes minimum and the estimates fulfil the constraints

$$\boldsymbol{g}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}) = \mathbf{0}\,. \tag{7.12}$$

Observe,

- the optimization function (7.11) does not depend on the variance factor σ_0^2.

[1] The Gauss–Markov model $\mathbb{E}(\underline{\boldsymbol{l}}) = \boldsymbol{f}(\boldsymbol{x})$ therefore can be interpreted as a special case of the Gauss–Helmert model, setting $\boldsymbol{g}(\boldsymbol{x}, \mathbb{E}(\underline{\boldsymbol{l}})) = -\mathbb{E}(\underline{\boldsymbol{l}}) + \boldsymbol{f}(\boldsymbol{x})$

[2] The definition of the implicit function is different from Förstner and Wrobel (2016, Eq. (4.426)), where the two arguments of the implicit function $\boldsymbol{g}$ are exchanged

[3] We us the variable $\boldsymbol{y}$ for the mean observation, in order to avoid to define approximate values for the fitted observations $\boldsymbol{y}$ within the iteration loop, since the intermediate values in an iteration scheme have no statistical meaning.

- In order to simplify the notation, and avoid statistical terms within the optimization procedure as far as possible, we will also write the optimization problem as follows:[4] For given observations $\boldsymbol{l}$, constraints $\boldsymbol{g}$ and weight matrix $W_{ll} = \Sigma_{ll}^{-1}$ find values for $\boldsymbol{x}$ and $\boldsymbol{y}$ that

$$\begin{aligned} &\text{minimize} && (\boldsymbol{y}-\boldsymbol{l})^{\mathsf{T}} W_{ll} (\boldsymbol{y}-\boldsymbol{l}) && (7.13)\\ &\text{subject to} && \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}\,, && (7.14) \end{aligned}$$

  where $\boldsymbol{y}$ stands for the unknown mean observation $\mathbb{E}(\underline{\boldsymbol{l}})$.

Remark: We may assume the observations appear in I statistically independent groups $\{\boldsymbol{l}_i, \Sigma_{l_i l_i}\}, i = 1, ..., I$, and if the dimension of these groups is the same, say d, we have $N = dI$. Furthermore, we often face the situation, that the constraints only refer to one group of observations. Then the functional model (7.12) can be written as

$$\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{y}_i) = \boldsymbol{0}\,, \quad i = 1, ..., I\,. \tag{7.15}$$

Hence, if the number of constraints per group is constant, say c, then the number of constraints is $G = cI$. As an example, this situation holds for the model of a 3D similarity for two sets of 3D points, where we have groups of $d = 6$ observations, namely the 3D coordinates in the two systems, and $c = 3$ constraints per group relating these coordinates via a similarity transformation with their parameters $\boldsymbol{x}$. ◇

We first provide a solution for the linear Gauss–Helmert model. We specialize it for independent and identically distributed observations and derive the solution for the two basic models, namely the Gauss–Markov model and the model with constraints between observations only. We also show, that the Gauss–Helmert model can be solved by chosing adequate substitute observations leading to a Gauss–Markov model. Since in case the model is non-linear the coefficient matrices need to be updated during the iteration process, why this model is called a *quasi Gauss–Markov model*. In the next section we then handle the non-linear case. Finally, we provide a derivation via an equivalent Gauss–Markov model.

7.4 The solutions for linear models

7.4.1 The solution for the basic linear Gauss–Helmert model

We start with the linear Gauss–Helmert model with covariance matrix $\mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll} = I$. We handle it as an algebraic, not a statistical optimization problem.

The original optimization problem reads as: for given observations $\boldsymbol{l} \in \mathbb{R}^N$, a regular $N \times N$ covariance matrix $\Sigma_{ll} = I_N$, full rank coefficient matrices. $A \in \mathbb{R}^{G \times U}$ and $Y \in \mathbb{R}^{G \times N}$ and a constant vector $\boldsymbol{b} \in \mathbb{R}^G$

$$\boxed{\begin{aligned} \text{GHM:}\quad &\text{minimize} && (\boldsymbol{y}-\boldsymbol{l})^{\mathsf{T}}(\boldsymbol{y}-\boldsymbol{l}) \\ &\text{subject to} && A\boldsymbol{x} + Y^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b} = \boldsymbol{0}\,. \end{aligned}} \tag{7.16}$$

w.r.t. the unknown parameters $\boldsymbol{x}$ and the mean observations $\boldsymbol{y}$.

Hence, here we chose the constraint function

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) = A\boldsymbol{x} + Y^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b} \tag{7.17}$$

which is linear in the unknown parameters. The coefficient matrices often are called *design matrices*, since they specify the design of the observation process. They are assumed to be given and fixed.

Furthermore, for a compact representation of the solution we use the *substituted observations* $\boldsymbol{n}(\boldsymbol{l})$ together with their covariance matrix

$$\boldsymbol{n}(\boldsymbol{l}) = Y^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b} \quad \text{and} \quad \mathbb{D}(\underline{\boldsymbol{n}}) = \Sigma_{nn} = Y^{\mathsf{T}}Y\,. \tag{7.18}$$

[4] This in the flavour of the problems discussed in Boyd and Vandenberghe (2004).

We obtain the estimated parameters and the fitted observations from

$$\boxed{\begin{array}{l}\widehat{\boldsymbol{x}} = \; -(\mathsf{X}^{\mathsf{T}}\Sigma_{nn}^{-1}\mathsf{X})^{-1}\,\mathsf{X}^{\mathsf{T}}\Sigma_{nn}^{-1}\,\boldsymbol{n}(\boldsymbol{l}) \\ \widehat{\boldsymbol{y}} = \boldsymbol{l} - \mathsf{Y}\Sigma_{nn}^{-1}\,\boldsymbol{g}(\widehat{\boldsymbol{x}},\boldsymbol{l})\,.\end{array}} \tag{7.19}$$

Remark: Generally, the parameters are estimated based on the normal equations

$$(\mathsf{X}^{\mathsf{T}}\Sigma_{nn}^{-1}\mathsf{X})\widehat{\boldsymbol{x}} + \mathsf{X}^{\mathsf{T}}\Sigma_{nn}^{-1}\,\boldsymbol{n}(\boldsymbol{l}) = \mathbf{0} \tag{7.20}$$

which can be solved in any numerical manner, especially if we want ot exploit the sparsity of X, Σ_{ll}, or Σ_{nn}. ◇

Proof: Using Lagrangian multipliers we need to find the minimum of

$$\boldsymbol{\Phi}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda}) = \frac{1}{2}(\boldsymbol{y}-\boldsymbol{l})^{\mathsf{T}}(\boldsymbol{y}-\boldsymbol{l}) + \boldsymbol{\lambda}^{\mathsf{T}}(\mathsf{X}\boldsymbol{x} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b})\,. \tag{7.21}$$

Necessary conditions are

$$\mathbf{0} = \frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{x}^{\mathsf{T}}} \;=\; \mathsf{X}^{\mathsf{T}}\boldsymbol{\lambda} \tag{7.22}$$

$$\mathbf{0} = \frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{y}^{\mathsf{T}}} \;=\; \boldsymbol{y} - \boldsymbol{l} + \mathsf{Y}\boldsymbol{\lambda} \tag{7.23}$$

$$\mathbf{0} = \frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{\lambda}^{\mathsf{T}}} \;=\; \mathsf{X}\boldsymbol{x} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b}\,. \tag{7.24}$$

Multiplying (7.23) with Y^{T} from the left leads to

$$\boldsymbol{y} = \boldsymbol{l} - \mathsf{Y}\boldsymbol{\lambda}\,. \tag{7.25}$$

Substituting this expression for $\boldsymbol{y}$ in (7.24) yields

$$\mathbf{0} = \mathsf{X}\boldsymbol{x} + \mathsf{Y}^{\mathsf{T}}(\boldsymbol{l} - \mathsf{Y}\boldsymbol{\lambda}) + \boldsymbol{b}\,, \tag{7.26}$$

which allows to solve for $\boldsymbol{\lambda}$

$$\boldsymbol{\lambda} = (\mathsf{Y}^{\mathsf{T}}\mathsf{Y})^{-1}(\mathsf{X}\boldsymbol{x} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})\,. \tag{7.27}$$

From (7.22) and (7.27) we obtain the normal equations for the estimates of the unknown parameters[5] $\boldsymbol{x}$

$$\mathsf{X}(\mathsf{Y}^{\mathsf{T}}\mathsf{Y})^{-1}\mathsf{X}\widehat{\boldsymbol{x}} = -\mathsf{X}(\mathsf{Y}^{\mathsf{T}}\mathsf{Y})^{-1}(\mathsf{Y}^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})\,. \tag{7.28}$$

From (7.25) and (7.27) we finally obtain estimates $\widehat{\boldsymbol{y}}$ for the mean observations[6] $\boldsymbol{y}$,

$$\widehat{\boldsymbol{y}} = \boldsymbol{l} - \mathsf{Y}(\mathsf{Y}^{\mathsf{T}}\mathsf{Y})^{-1}(\mathsf{X}\widehat{\boldsymbol{x}} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})\,, \tag{7.29}$$

as a function of the estimated parameters $\widehat{\boldsymbol{x}}$ and the observations $\boldsymbol{l}$.

7.4.2 The Gauss–Helmert model for general covariance matrix

The Gauss–Helmert model with general covariance matrix reads as: for given observations $\boldsymbol{l}$, regular covariance matrix $\Sigma_{ll} = W_{ll}^{-1}$, and coefficient matrices X and Y

$$\boxed{\begin{array}{lll}\text{GHM}(\Sigma): & \text{minimize} & (\boldsymbol{y}-\boldsymbol{l})^{\mathsf{T}}\Sigma_{ll}^{-1}(\boldsymbol{y}-\boldsymbol{l}) \\ & \text{subject to} & \mathsf{X}\boldsymbol{x} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b} = \mathbf{0}\,.\end{array}} \tag{7.30}$$

[5] Observe, for given substitute observations $\boldsymbol{n} = \mathsf{Y}^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b}$, this is the solution for the Gauss–Markov model minimizing $(\mathsf{X}\boldsymbol{x}+\boldsymbol{n})\Sigma_{nn}^{-1}(\mathsf{X}\boldsymbol{x}+\boldsymbol{n})$ w.r.t. the parameters $\boldsymbol{x}$

[6] Observe, for fixed $\widehat{\boldsymbol{x}}$, this is the solution of the problem with constraints for observations $\boldsymbol{l}$ only, minimizing $|\mathsf{X}\widehat{\boldsymbol{x}} + \mathsf{Y}^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})|_2$ w.r.t. the observations $\boldsymbol{l}$, leading to fitted observations $\widehat{\boldsymbol{l}} = \widehat{\boldsymbol{y}}$.

w.r.t. the unknown parameters $\boldsymbol{x}$ and the mean observations $\boldsymbol{y}$. For a compact representation of the solution we use the substituted observations with their – now different – covariance matrix

$$\boldsymbol{n} = Y\boldsymbol{l} + \boldsymbol{b} \quad \text{and} \quad \mathbb{D}(\underline{\boldsymbol{n}}) = \Sigma_{nn} = Y^{\mathsf{T}}\Sigma_{ll}Y\,. \tag{7.31}$$

We obtain the estimated parameters and the fitted observations from

$$\boxed{\begin{array}{l}\hat{\boldsymbol{x}} = \ -(X^{\mathsf{T}}\Sigma_{nn}^{-1}X)^{-1}\,X^{\mathsf{T}}\Sigma_{nn}^{-1}\,\boldsymbol{n}(\boldsymbol{l}) \\ \hat{\boldsymbol{y}} = \boldsymbol{l} - \Sigma_{ll}Y\Sigma_{nn}^{-1}\,\boldsymbol{g}(\hat{\boldsymbol{x}},\boldsymbol{l})\,.\end{array}} \tag{7.32}$$

Proof: We transfer this model to an unweighted Gauss–Helmert model. Especially, we eliminate the weights of the observations. For eliminating the weights, we use the substitutions

$$Y_g = \Sigma_{ll}^{1/2}Y \ , \quad \boldsymbol{l}_g = \Sigma_{ll}^{-1/2}\,\boldsymbol{l}\,, \quad \text{and} \quad \boldsymbol{y}_g = \Sigma_{ll}^{-1/2}\,\boldsymbol{y}\,. \tag{7.33}$$

Now, we need to solve the following unweighted Gauss–Helmert model: for given observations $\boldsymbol{l}_g$ and coefficient matrices X and Y_g,

$$\begin{array}{lll}\text{GHM(w):} & \text{minimize} & (\boldsymbol{y}_g - \boldsymbol{l}_g)^{\mathsf{T}}(\boldsymbol{y}_g - \boldsymbol{l}_g) \\ & \text{subject to} & X\boldsymbol{x} + Y_g^{\mathsf{T}}\boldsymbol{y}_g = \boldsymbol{0}\,,\end{array} \tag{7.34}$$

w.r.t. the unknown parameters $\boldsymbol{x}$ and the mean observations $\boldsymbol{y}_g$.

We thus obtain the normal equation system

$$X(Y_g^{\mathsf{T}}Y_g)^{-1}X\hat{\boldsymbol{x}} = -X(Y_g^{\mathsf{T}}Y_g)^{-1}(Y_g^{\mathsf{T}}\boldsymbol{l}_g + \boldsymbol{b})\,. \tag{7.35}$$

or explicitly

$$X^{\mathsf{T}}(Y^{\mathsf{T}}\Sigma_{ll}Y)^{-1}X\ \hat{\boldsymbol{x}} = -X^{\mathsf{T}}(Y^{\mathsf{T}}\Sigma_{ll}Y)^{-1}Y^{\mathsf{T}}\boldsymbol{l} \tag{7.36}$$

The fitted observations we obtain from

$$\hat{\boldsymbol{y}}_g = \boldsymbol{l}_g - Y_g(Y_g^{\mathsf{T}}Y_g)^{-1}(X\hat{\boldsymbol{x}} + Y_g^{\mathsf{T}}\boldsymbol{l}_g + \boldsymbol{b}))\,. \tag{7.37}$$

or finally

$$\hat{\boldsymbol{y}} = \boldsymbol{l} - \Sigma_{ll}Y(Y^{\mathsf{T}}\Sigma_{ll}Y)^{-1}(X\hat{\boldsymbol{x}} + Y^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})\,. \tag{7.38}$$

7.4.3 Gauss–Markov model

The Gauss–Markov results from specializing the design matrix Y in the Gauss–Helmert model to

$$Y = -I\,, \tag{7.39}$$

leading to the constraint function

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = X\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{b} \tag{7.40}$$

and substitute observations and their covariance matrix

$$\boldsymbol{n}(\boldsymbol{l}) = -\boldsymbol{l} + \boldsymbol{b} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{n}}) = \Sigma_{nn} = \Sigma_{ll}\,. \tag{7.41}$$

The Gauss–Markov model with covariance matrix $\Sigma_{ll} = W_{ll}^{-1}$ leads to the following general least squares optimization problem, for given observations $\boldsymbol{l}$, weight matrix $W_{ll} = \Sigma_{ll}^{-1}$, and coefficient matrix X

$$\boxed{\begin{array}{lll}\text{GMM}(\Sigma)\text{:} & \text{minimize} & (\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}W_{ll}(\boldsymbol{y} - \boldsymbol{l}) \\ & \text{subject to} & \boldsymbol{y} = X\boldsymbol{x} + \boldsymbol{b}\,.\end{array}} \tag{7.42}$$

w.r.t. the unknown parameters $\boldsymbol{x}$ and mean observations $\boldsymbol{y}$. It yields the optimal parameters

$$\boxed{\begin{array}{l}\hat{\boldsymbol{x}} = -(X^{\mathsf{T}}W_{ll}X)^{-1}X^{\mathsf{T}}W_{ll}\ \boldsymbol{n}(\boldsymbol{l}) \\ \hat{\boldsymbol{y}} = \boldsymbol{l} - \boldsymbol{g}(\hat{\boldsymbol{x}},\boldsymbol{l})\end{array}} \tag{7.43}$$

or explicitly in the classical form

$$\begin{array}{l}\hat{\boldsymbol{x}} = (X^{\mathsf{T}}W_{ll}X)^{-1}X^{\mathsf{T}}W_{ll}(\boldsymbol{l} - \boldsymbol{b}) \\ \hat{\boldsymbol{y}} = X\hat{\boldsymbol{x}} + \boldsymbol{b}\,.\end{array} \tag{7.44}$$

7.4.4 Model with constraints between the observations only

The model with constraints between the observations only results from specializing the design matrix A in the Gauss–Helmert model to:

$$A = 0\,, \tag{7.45}$$

leading to the constraint function

$$\boldsymbol{g}(\boldsymbol{y}) = A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b}\,, \tag{7.46}$$

not depending on unknown parameters $\boldsymbol{x}$, and substitute observations and their covariance matrix

$$\boldsymbol{n}(\boldsymbol{l}) = A^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{n}}) = \Sigma_{nn} = A^{\mathsf{T}}\Sigma_{ll}A^{\mathsf{T}}\,. \tag{7.47}$$

The model with constraints between the given observations $\boldsymbol{l}$ having covariance matrix Σ_{ll} leads to the following least squares problem

$$\boxed{\begin{array}{lll} \text{CONSTR}(\Sigma): & \text{minimize} & (\boldsymbol{y}-\boldsymbol{l})^{\mathsf{T}}\Sigma_{ll}^{-1}(\boldsymbol{y}-\boldsymbol{l}) \\ & \text{subject to} & A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b} = \boldsymbol{0}\,. \end{array}} \tag{7.48}$$

w.r.t. the mean observations $\boldsymbol{y}$. It yields the optimal estimates for the fitted observations

$$\boxed{\widehat{\boldsymbol{y}} = \boldsymbol{l} - \Sigma_{ll}A(A^{\mathsf{T}}\Sigma_{ll}A^{\mathsf{T}})^{-1}\,(A^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b})\,.} \tag{7.49}$$

7.4.5 The quasi Gauss–Markov model

As already indicated in the footnotes for (7.28) and (7.29) we can perform the estimation in the Gauss–Helmert model in two steps:

1. First we perform a Gauss–Markov model using the substitute observations

   $$\boldsymbol{n} = A^{\mathsf{T}} + \boldsymbol{b} \tag{7.50}$$

   hence

   $$\boldsymbol{n} = X\boldsymbol{x} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{n}}) = A^{\mathsf{T}}\Sigma_{ll}A\,, \tag{7.51}$$

   Using (7.43), this leads to the optimal estimates for the parameters $\boldsymbol{x}$ using the normal equations

   $$\widehat{\boldsymbol{x}} = (X^{\mathsf{T}}W_{nn}X)^{-1}\,X^{\mathsf{T}}W_{nn}\,\boldsymbol{n}\,. \tag{7.52}$$

2. Now, as we have the optimal estimates $\widehat{\boldsymbol{x}}$, we can treat them as fixed values. With the constant vector

   $$\boldsymbol{c}(\widehat{\boldsymbol{x}}) = X\widehat{\boldsymbol{x}} + \boldsymbol{b}\,, \tag{7.53}$$

   thus

   $$\boldsymbol{g}(\boldsymbol{l}) = A^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{c}(\widehat{\boldsymbol{x}})\,, \tag{7.54}$$

   we can find the estimates for the fitted observations from the model for constraints between the observations only

   $$A^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{c}(\widehat{\boldsymbol{x}}) = \boldsymbol{0} \quad \text{and} \quad \mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll}\,. \tag{7.55}$$

   With (7.49), this leads to the estimates

   $$\widehat{\boldsymbol{y}} = \boldsymbol{l} - \Sigma_{ll}A(A^{\mathsf{T}}\Sigma_{ll}A)^{-1}\,\boldsymbol{g}(\boldsymbol{l}) \tag{7.56}$$

The Gauss–Markov model (7.51) is called the *quasi Gauss–Markov model* in the context of solving the parameters in the Gauss–Helmert model. In case the constraints are non-linear, the coefficient matrices are not fixed but need to be updated during the iteration process, which motivates the prefix *quasi*.

7.4.6 Results using pseudo inverses

The results can be written compactly using pseudo inverses. This is motivated from the least-squares solution of the simple Gauss–Markov model relating the mean observations to the unknown parameters via

$$\boldsymbol{y} = X\boldsymbol{x} \tag{7.57}$$

and minimizing $|\boldsymbol{y} - \boldsymbol{l}|_2$. This leads to the classical solution $\hat{\boldsymbol{x}} = (X^\mathsf{T}X)^{-1}X^\mathsf{T}\ \boldsymbol{l}$, which with the pseudo inverse

$$X^+ = (X^\mathsf{T}X)^{-1}X^\mathsf{T} \tag{7.58}$$

can be written as

$$\hat{\boldsymbol{x}} = X^+\boldsymbol{l} \tag{7.59}$$

This is an intuitive description of the inversion of (7.57), keeping in mind, that the inversion is not unique, since X is not regular, and regularization is enforced by the least squares principle.

Similarly, in case we minimize a weighted sum of squares $(\boldsymbol{y} - \boldsymbol{l})^\mathsf{T}W(\boldsymbol{y} - \boldsymbol{l})$ w.r.t. the parameters $\boldsymbol{x}$, with the weighted pseudo inverse

$$X^+_{\boldsymbol{w}} = (X^\mathsf{T}WX)^{-1}X^\mathsf{T}W \tag{7.60}$$

we obtain the solution

$$\hat{\boldsymbol{x}} = X^+_{\boldsymbol{w}}\boldsymbol{l}\,. \tag{7.61}$$

We first define the properties of pseudo inverses and then provide the solutions of the different estimation problems.

7.4.6.1 Pseudo inverse and weighted pseudo inverse

For the regular $M \times N$ matrix A, with $M \geq N$ and $\mathrm{rk}(A) = N$ we use the pseudo inverse A^+:

$$A^+ := (A^\mathsf{T}A)^{-1}A^\mathsf{T} \tag{7.62}$$

It fulfils further the four relations:

$$AA^+A = A \quad A^+AA^+ = A^+ \quad (AA^+)^\mathsf{T} = AA^+ \quad A^+A = I\,. \tag{7.63}$$

Similarly, with the symmetric weight matrix U we use the weighted pseudo inverse (see Pepić (2010))

$$A^+_U := (A^\mathsf{T}UA)^{-1}A^\mathsf{T}U \tag{7.64}$$

which fulfils the four relations

$$AA^+_UA = A \quad A^+_UAA^+_U = A^+_U \quad (UAA^+_U)^\mathsf{T} = UAA^+ \quad A^+_UA = I\,. \tag{7.65}$$

7.4.6.2 Solutions with pseudo inverses

We explicitly use the following inverses:

$$\begin{aligned}
X^+ &= (X^\mathsf{T}X)^{-1}\ X^\mathsf{T} && (7.66)\\
X^+_{\boldsymbol{w}_{ll}} &= (X^\mathsf{T}W_{ll}X)^{-1}\ X^\mathsf{T}W_{ll} && (7.67)\\
X^+_{\boldsymbol{w}_{nn}} &= (X^\mathsf{T}W_{nn}X)^{-1}\ X^\mathsf{T}W_{nn} && (7.68)\\
Y^+ &= (Y^\mathsf{T}Y)^{-1}\ Y^\mathsf{T} && (7.69)\\
Y^+_{\Sigma_{ll}} &= (Y^\mathsf{T}\Sigma_{ll}Y)^{-1}\ Y^\mathsf{T}\Sigma_{ll} && (7.70)
\end{aligned}$$

Then we obtain the following solutions:

- Gauss–Markov model ($Y = -I$). Starting from the model

$$\boldsymbol{y} - \boldsymbol{b} = X\ \boldsymbol{x} \tag{7.71}$$

we obtain

$$\begin{aligned} \hat{\boldsymbol{x}} &= -X^{+}_{\boldsymbol{w}_{ll}}\ \boldsymbol{n}(\boldsymbol{l}) \qquad \text{and} \qquad \hat{\boldsymbol{y}} &= \boldsymbol{l} + \boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l}) \quad (7.72)\\ &= X^{+}_{\boldsymbol{w}_{ll}}\ (\boldsymbol{l} - \boldsymbol{b}) & = X\ \hat{\boldsymbol{x}} + \boldsymbol{b}\,. \quad (7.73) \end{aligned}$$

- Model with constraints between the observations only ($X = 0$). Starting from the model

$$Y^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{l}) + \boldsymbol{g}(\boldsymbol{l}) = \boldsymbol{0} \tag{7.74}$$

we arrive at the solution $\hat{\boldsymbol{y}} - \boldsymbol{l} = -Y^{+\mathsf{T}}_{\Sigma_{ll}}\ \boldsymbol{g}(\boldsymbol{l})$, or

$$\hat{\boldsymbol{y}} \quad = \quad \boldsymbol{l} - Y^{+\mathsf{T}}_{\Sigma_{ll}}\ \boldsymbol{g}(\boldsymbol{l}) \tag{7.75}$$

- Gauss–Helmert model. Starting from the model

$$X\boldsymbol{x} + \underbrace{Y^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b}}_{\boldsymbol{n}(\boldsymbol{y})} = Y^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{l}) + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \boldsymbol{0} \tag{7.76}$$

when first using $\boldsymbol{n}(\boldsymbol{l})$ as observations and then fixing the estimate for $\boldsymbol{x}$ we arrive at

$$\hat{\boldsymbol{x}} = -X^{+}_{\boldsymbol{w}_{nn}}\ \boldsymbol{n}(\boldsymbol{l}) \qquad \text{and} \qquad \hat{\boldsymbol{y}} \quad = \quad \boldsymbol{l} - Y^{+\mathsf{T}}_{\Sigma_{ll}}\ \boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l}) \tag{7.77}$$

taking the covariance matrix Σ_{nn} of $\underline{\boldsymbol{n}}(\underline{\boldsymbol{l}})$ into account.

The solutions are collected in the following Table, starting with the Gauss–Helmert model with general covariance matrix and then showing the different specializations.

Table 7.1: Statistically optimal solutions in the linear model $(X, Y, \mathbb{D}(\underline{\boldsymbol{l}}))$ with its specializations: $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) = X\boldsymbol{x} + Y^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b} = \boldsymbol{0}$ relating the mean $\boldsymbol{y} = \mathbb{E}(\underline{\boldsymbol{l}})$ of the observations $\underline{\boldsymbol{l}}$ to the unknown parameters $\boldsymbol{x}$ assuming a general covariance matrix and a unit matrix $\mathbb{D}(\underline{\boldsymbol{l}}) = \Sigma_{ll}$ and $\mathbb{D}(\underline{\boldsymbol{l}}) = I$, respectively. We use the substitute observations $\boldsymbol{n}(\boldsymbol{l}) = Y^{\mathsf{T}}\boldsymbol{l} + \boldsymbol{b}$ with their covariance matrix Σ_{nn}.
Rows 1 and 2: Gauss–Helmert model.
Rows 3 and 4: Gauss–Markov: $\boldsymbol{n}(\boldsymbol{l}) = -\boldsymbol{l} + \boldsymbol{b}$.
Rows 5 and 6: Model with constraints between the observations: $\boldsymbol{g}(\boldsymbol{y}) = Y^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{b}$.

	model$(X, Y, \mathbb{D}(\underline{\boldsymbol{l}}))$	task		solution		
1	GHM(X, Y, Σ_{ll})	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}\Sigma_{ll}^{-1}(\boldsymbol{y} - \boldsymbol{l})$	$\hat{\boldsymbol{x}} =$	$-\ X^{+}_{\boldsymbol{w}_{nn}}$	$\boldsymbol{n}(\boldsymbol{l})$
		s.t.	$X\boldsymbol{x} + Y^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$	$Y^{+\mathsf{T}}_{\Sigma_{ll}}$	$\boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l})$
2	GHM(X, Y, I)	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{l})$	$\hat{\boldsymbol{x}} =$	$-\ X^{+}_{\boldsymbol{w}_{nn}}$	$\boldsymbol{n}(\boldsymbol{l})$
		s.t.	$X\boldsymbol{x} + Y^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$	$Y^{+\mathsf{T}}$	$\boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l})$
3	GMM$(X, -I, \Sigma_{ll})$	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}\Sigma_{ll}^{-1}(\boldsymbol{y} - \boldsymbol{l})$	$\hat{\boldsymbol{x}} =$	$-\ X^{+}_{\boldsymbol{w}_{ll}}$	$\boldsymbol{n}(\boldsymbol{l})$
		s.t.	$X\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$		$\boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l})$ [1)]
4	GMM$(X, -I, I)$	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{l})$	$\hat{\boldsymbol{x}} =$	$-\ X^{+}$	$\boldsymbol{n}(\boldsymbol{l})$
		s.t.	$X\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$		$\boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{l})$ [1)]
5	CONSTR$(0, Y, \Sigma_{ll})$	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}\Sigma_{ll}^{-1}(\boldsymbol{y} - \boldsymbol{l})$			
		s.t.	$Y^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$	$Y^{+\mathsf{T}}_{\Sigma_{ll}}$	$\boldsymbol{g}(\boldsymbol{l})$
6	CONSTR$(0, Y, I)$	min.	$(\boldsymbol{y} - \boldsymbol{l})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{l})$			
		s.t.	$Y^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{c}$	$\hat{\boldsymbol{y}} = \boldsymbol{l} -$	$Y^{+\mathsf{T}}$	$\boldsymbol{g}(\boldsymbol{l})$

[1)] This is equivalent to $\hat{\boldsymbol{y}} = X\hat{\boldsymbol{x}} + \boldsymbol{b}$

This closes the section on the estimation in the linear Gauss–Helmert model. We did not construct the solutions, but just proved they are correct. The generalization to non-linear constraints will also use the reduction to a Gauss–Markov model, but derive the iterative solution explicitly. Moreover, coefficient matrices A and B then depend on the current estimates of the parameters and the observations thus need to be updated in each iteration.

7.5 The non-linear Gauss–Helmert model

The functional model generally is non-linear. We assume we have approximate values $\boldsymbol{x}^a$ and $\boldsymbol{y}^a$ for the parameters $\boldsymbol{x}$ and the mean observations $\boldsymbol{y}$ and updates

$$\boldsymbol{x} := u_x(\boldsymbol{x}^a, \Delta\boldsymbol{x})) \quad \text{e.g.,} \quad \boldsymbol{x}^a := \boldsymbol{x}^a + \Delta\boldsymbol{x}\,. \tag{7.78}$$

and

$$\boldsymbol{y} := u_y(\boldsymbol{y}^a, \Delta\boldsymbol{y}) \quad \text{e.g.,} \quad \boldsymbol{y}^a := \boldsymbol{y}^a + \Delta\boldsymbol{y}\,. \tag{7.79}$$

These relations hold for small corrections $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{y}$. Given values for $\boldsymbol{x}$ and its approximations $\boldsymbol{x}^a$ we assume we can determine the corrections from

$$\Delta\boldsymbol{x} = u_x^{-1}(\boldsymbol{x}, \boldsymbol{x}^a) \quad \text{e.g.,} \quad \Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^a \tag{7.80}$$

Similarly, we assume there exist inverse functions for the mean observations

$$\Delta\boldsymbol{y} = u_y^{-1}(\boldsymbol{y}, \boldsymbol{y}^a) \quad \text{e.g.,} \quad \Delta\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}^a \tag{7.81}$$

Hence we have the update function with $m \geq n$, for small $\Delta\boldsymbol{x}$, especially for $m = n$

$$u_x : \mathbb{R}^n \mapsto \mathbb{R}^m \quad \Delta\boldsymbol{x} \mapsto \boldsymbol{x} = u_x(\Delta\boldsymbol{x}; \boldsymbol{x}^a) \quad \text{especially} \quad \boldsymbol{x} = \Delta\boldsymbol{x} + \boldsymbol{x}^a \tag{7.82}$$

$$u_x^{-1} : \mathbb{R}^m \mapsto \mathbb{R}^n \quad \boldsymbol{x} \mapsto \Delta\boldsymbol{x} = u_x^{-1}(\boldsymbol{x}; \boldsymbol{x}^a) \quad \text{especially} \quad \Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^a\,, \tag{7.83}$$

and similarly, for u_y.

Example: Non-linear update and its inversion for 3D rotations. Let the unknown parameters be a 3×3 rotation matrix R. We actually estimate a small 3-vector $\Delta\boldsymbol{r}$ of small rotation angles. The approximate rotation matrix R^a the can be corrected using

$$R = u_x(R^a, \Delta\boldsymbol{r}) = R(\Delta\boldsymbol{r})\, R^a\,. \tag{7.84}$$

where $R(\Delta\boldsymbol{r})$ is a rotation matrix depending on the 3-vector $\Delta\boldsymbol{r}$, e.g., using the exponential or the Cayley form

$$R(\Delta\boldsymbol{r}) = \exp(S(\Delta\boldsymbol{r})) \quad \text{or} \quad R(\Delta\boldsymbol{r}) = (I + S(\Delta\boldsymbol{r}/2))(I - S(\Delta\boldsymbol{r}/2))^{-1} \tag{7.85}$$

with the skew symmetric matrix $S(\boldsymbol{a})$ inducing the cross product $\boldsymbol{a} \times \boldsymbol{b} = S(\boldsymbol{a})\boldsymbol{b}$. In case we have given R and some approximation R^a, we may determine the correction vector $\Delta\boldsymbol{r}$ from

$$S(\Delta\boldsymbol{r}) = \log(R^\mathsf{T} R^a) \approx R^\mathsf{T} R^a - I\,, \tag{7.86}$$

thus taking the off diagonal terms of the product $R^\mathsf{T} R^a$ of the two rotation matrices as the sought 3-vector. This can compactly be written as

$$\Delta\boldsymbol{r} = u_x^{-1}(R, R^a) = s(R^\mathsf{T} R^a)\,. \tag{7.87}$$

where the function

$$s(A) = \frac{1}{2}\begin{bmatrix} A_{32} - A_{23} \\ A_{13} - A_{31} \\ A_{21} - A_{21} \end{bmatrix} \tag{7.88}$$

extracts the skew vector of the 3×3 rotation matrix A. ◇

Similarly, we have the updates and their inversion starting from $\boldsymbol{l}$, first for the approximations of the mean observations

$$\boldsymbol{y}^a = u_y(\boldsymbol{l}, \boldsymbol{v}^a) \quad \text{and} \quad \boldsymbol{v}^a = u_y^{-1}(\boldsymbol{y}^a, \boldsymbol{l}) = -u_y^{-1}(\boldsymbol{l}, \boldsymbol{y}^a)\,. \tag{7.89}$$

which for small residuals can be defined in either manner. Thus we have for the mean observations

$$\boldsymbol{y} = u_y(\boldsymbol{l}, \boldsymbol{v}) \quad \text{and} \quad \boldsymbol{v} = u_y^{-1}(\boldsymbol{y}, \boldsymbol{l}) = -u_y^{-1}(\boldsymbol{l}, \boldsymbol{y})\,. \tag{7.90}$$

For small values we have

$$\boldsymbol{v} = \boldsymbol{v}^a + \Delta\boldsymbol{y}\,, \tag{7.91}$$

see Fig. 7.1. Since the observations $\boldsymbol{l}$ and the residuals $\boldsymbol{v}$ may have a different structure, e.g., if the observations are rotation matrices and the residuals are rotation vectors, the covariance matrix Σ_{ll} refers to the residuals of the observations

Covariance matrix for rotation matrices. In the case of an observed rotation matrix R, we represent the uncertain rotation as

$$\underline{R} = R(\underline{\boldsymbol{r}})\,\mathbb{E}(\underline{R}) \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{r}}) = \Sigma_{rr} \tag{7.92}$$

If R is observed, the we refer to the 3×3 matrix Σ_{rr} as the covariance matrix Σ_{ll} of the observed rotation. ◇

We are now prepared to derive a linear substitute problem used for iteratively determining the unknowns $\boldsymbol{y}$ and $\boldsymbol{x}$.

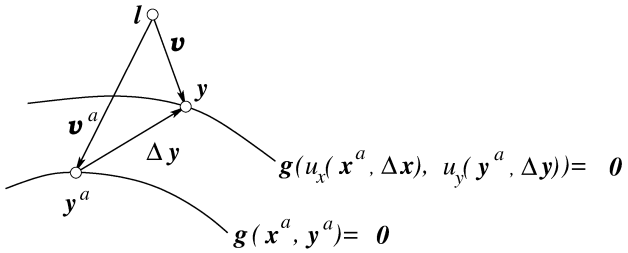


Figure 7.1: Update of the unknowns and the mean observations in the Gauss–Helmert model. The corrections $\Delta\boldsymbol{x} = u_x^{-1}(\boldsymbol{x}, \boldsymbol{x}^a)$ to the parameters and the corrections $\Delta\boldsymbol{y} = u_y^{-1}(\boldsymbol{y}, \boldsymbol{y}^a) = \boldsymbol{v} - \boldsymbol{v}^a$ to the mean observations and residuals are meant to converge to zero. The figure assumes the dimensions of the observations/parameters $(\boldsymbol{l}, \boldsymbol{y}, \boldsymbol{y}^a)$ and the dimensions of their residuals/corrections $(\boldsymbol{v}, \boldsymbol{v}^a, \boldsymbol{y}^a)$ are the same

1. We define the corrections to the parameters and the mean observations

   corrections to mean observations and parameters

   $$\Delta\boldsymbol{x} = u_x^{-1}(\boldsymbol{x}, \boldsymbol{x}^a) = \boldsymbol{x}^a - \boldsymbol{x} \quad \text{and} \quad \Delta\boldsymbol{y} = u_y^{-1}(\boldsymbol{y}, \boldsymbol{y}^a) = \boldsymbol{v} - \boldsymbol{v}^a\,, \tag{7.93}$$

   in order to iteratively improve the approximations $\boldsymbol{x}^a$ and $\boldsymbol{y}^a$ such that after convergence $\Delta\boldsymbol{x} = \boldsymbol{0}$ and $\Delta\boldsymbol{y} = \boldsymbol{0}$. Observe, that the approximate residuals $\boldsymbol{v}^a = \boldsymbol{y}^a - \boldsymbol{l}$ also are to be corrected by $\Delta\boldsymbol{y}$.

2. The optimization function then reads as[7]

   $$\boxed{\Omega = (\boldsymbol{v}^a + \Delta\boldsymbol{y})^\mathsf{T}\,\Sigma_{ll}^{-1}\,(\boldsymbol{v}^a + \Delta\boldsymbol{y})\,.} \tag{7.94}$$

   where the covariance matrix Σ_{ll} refers to the corrections $\boldsymbol{v}$ of the observations.

   Remark: Observe, the optimization function (7.94) of this non-linear model results from the one (7.30) of the linear model using $\boldsymbol{y} - \boldsymbol{l} = \boldsymbol{y}^a + \Delta\boldsymbol{y} - \boldsymbol{l} = \boldsymbol{v}^a + \Delta\boldsymbol{y}$. ◇

3. Linearization of the nonlinear implicit function (7.12) leads to the constraints *linearized constraints*

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{g}(u_x(\boldsymbol{x}^a,\Delta\boldsymbol{x}),u_y(\boldsymbol{y}^a,\Delta\boldsymbol{y})) = \boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{y}^a) + \mathsf{X}\Delta\boldsymbol{x} + \mathsf{Y}^\mathsf{T}\Delta\boldsymbol{y} = \boldsymbol{0} \tag{7.95}$$

with the Jacobians

$$\boxed{\underset{G\times U}{\mathsf{X}} = \left.\frac{\partial \boldsymbol{g}}{\partial \Delta\boldsymbol{x}}\right|_{x=x^a,y=y^a} \quad \text{and} \quad \underset{G\times N}{\mathsf{Y}^\mathsf{T}} = \left.\frac{\partial \boldsymbol{g}}{\partial \Delta\boldsymbol{y}}\right|_{x=x^a,y=y^a},} \tag{7.96}$$

to be evaluated at the approximations of the mean observations and of the parameters.

Remark: Also the structure of the constraints of the linear Gauss–Helmert model is preserved, when replacing the unknowns $\boldsymbol{x}$ and $\boldsymbol{y}$ by their corrections $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{y}$ and the constant $\boldsymbol{b}$ by $\boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{y}^a)$. ◇

Therefore the *linear substitute problem* for determining the corrections $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{y}$ is: *linear substitute problem*

$$\text{minimize} \quad (\boldsymbol{v}^a + \Delta\boldsymbol{y})^\mathsf{T}\,\Sigma_{ll}^{-1}\,(\boldsymbol{v}^a + \Delta\boldsymbol{y}) \tag{7.97}$$

$$\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{y}^a) + \mathsf{X}\Delta\boldsymbol{x} + \mathsf{Y}^\mathsf{T}\Delta\boldsymbol{y} = \boldsymbol{0}\,, \tag{7.98}$$

for given approximate values $\boldsymbol{y}^a$ and thus $\boldsymbol{v}^a = u_y^{-1}(\boldsymbol{y}^a,\boldsymbol{l})$, function $\boldsymbol{g}$, Jacobians X and Y, and covariance matrix Σ_{ll}.

We first will provide the algorithm and then its derivation.

7.5.1 The algorithm for estimating the parameters

We start from the observations $\{\boldsymbol{l},\Sigma_{ll}\}$, the implicit functions $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}$, and the approximate values $\boldsymbol{x}^a$ for the unknowns and $\boldsymbol{y}^a$ for the mean observations, which are initiated with $\boldsymbol{y}^a := \boldsymbol{l}$. We obtain the following algorithm for an iterative solution:

1. Iterate until convergence
   (a) Determine the Jacobians X and Y (7.96) at the current approximate values $(\boldsymbol{x}^a,\boldsymbol{y}^a)$. *Jacobians at current approximations*
   (b) Determine the contradictions $\boldsymbol{c}_g$ of the negative constraints at the appproximate values $\boldsymbol{x}^a$ of the unknown parameters together with their weight matrix [8] [9]

$$\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{l}) \quad \text{and} \quad \mathsf{W}_{gg} = (\mathsf{Y}^\mathsf{T}\Sigma_{ll}\mathsf{Y})^{-1}\,. \tag{7.99}$$

contradictions of constraints given the parameters

   (c) Solve the normal equation system for the corrections $\Delta\boldsymbol{x}$ of the parameters

$$\mathsf{N}\Delta\boldsymbol{x} = \boldsymbol{m} \quad \text{with} \quad \mathsf{N} = \mathsf{X}^\mathsf{T}\mathsf{W}_{gg}\mathsf{X} \quad \text{and} \quad \boldsymbol{m} = \mathsf{X}^\mathsf{T}\mathsf{W}_{gg}\,\boldsymbol{c}_g\,. \tag{7.100}$$

normal equation system

   (d) Update the approximate parameters

$$\boldsymbol{x}^a := u_x(\boldsymbol{x}^a,\Delta\boldsymbol{x}) \quad \text{e.g.,} \quad \boldsymbol{x}^a := \boldsymbol{x}^a + \Delta\boldsymbol{x}\,. \tag{7.101}$$

   hence

$$-\boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{l}) := \boldsymbol{c}_g - \mathsf{X}\Delta\boldsymbol{x} \tag{7.102}$$

   (e) Determine the corrections for the mean observations

$$\Delta\boldsymbol{y} = \Sigma_{ll}\mathsf{Y}\mathsf{W}_{gg}(\boldsymbol{c}_g - \mathsf{X}\Delta\boldsymbol{x}) - \boldsymbol{v}^a\,. \tag{7.103}$$

   (f) Update the approximate mean observations *update of approximate mean observations*

$$\boldsymbol{y}^a := u_y(\boldsymbol{y}^a,\Delta\boldsymbol{y}) \quad \text{e.g.,} \quad \boldsymbol{y}^a := \boldsymbol{y}^a + \Delta\boldsymbol{y}\,. \tag{7.104}$$

2. Set the final estimates of the unknown parameters and of the mean observations, sometimes called the fitted observation $\hat{\boldsymbol{l}} := \hat{\boldsymbol{y}}$ *final estimates*

$$\hat{\boldsymbol{x}} := \boldsymbol{x}^a \quad \text{and} \quad \hat{\boldsymbol{y}} = \boldsymbol{y}^a \,. \tag{7.105}$$

3. Determine the covariance matrix of the estimated parameters *covariance matrix of the estimated parameters*

$$\Sigma_{\hat{x}\hat{x}} = N^{-1} \,. \tag{7.106}$$

4. If we only know an approximate covariance matrix Σ_{ll}^a and we assume the covariance matrix Σ_{ll} differs from the approximation by an unknown variance factor σ_0^2

$$\Sigma_{ll} = \sigma_0^2 \Sigma_{ll}^a \quad \text{with} \quad W_{ll} = \Sigma_{ll} \,, \tag{7.107}$$

then we can perform the estimation with Σ_{ll}^a, instead of using Σ_{ll}, which has no effect onto the estimates. But then we can find an estimate

$$\hat{\sigma}_0^2 = \frac{\boldsymbol{c}_l^\mathsf{T} W_{ll}^a \boldsymbol{c}_l}{G-U} \quad \text{or} \quad \hat{\sigma}_0^2 = \frac{\boldsymbol{c}_g^\mathsf{T} W_{c_g c_g}^a \boldsymbol{c}_g}{G-U} \,. \tag{7.108}$$

for the estimated variance factor. Then we obtain an estimate for the covariance matrix of the estimated parameters *estimated variance factor*

$$\hat{\Sigma}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2 \Sigma_{\hat{x}\hat{x}}^a \quad \text{with} \quad \Sigma_{\hat{x}\hat{x}}^a = (X^\mathsf{T}(Y^\mathsf{T}\Sigma_{ll}^a Y)^{-1} X)^{-1} \,. \tag{7.109}$$

the attribute *estimated* only referring to use of the estimated variance factor.

Remark: If the observational noise is small and an approximate solution is acceptable, the steps 1.(e–f) can be omitted. Then the Jacobians X and Y are to be determined at $(\boldsymbol{x}^a, \boldsymbol{l})$ instead of at $(\boldsymbol{x}^a, \boldsymbol{y}^a)$. ◇

The complete procedure is given in the algorithm below. The green parts refer to the case, where the degrees of freedom of the parameters and observations is less than the number of elements of their representation.

7.5.2 Derivation of the procedure

We now derive the procedure.

7.5.2.1 Estimating the parameters with a quasi Gauss–Markov model

We start from the constraint (7.95) rewritten as

$$-\boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{y}^a) - Y^\mathsf{T} \Delta\boldsymbol{y} = X \Delta\boldsymbol{x} \,. \tag{7.110}$$

In order to eliminate the dependeny of $\Delta\boldsymbol{y}$, we introduce the *contradiction of the constraints*, i.e., the value

$$\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{l}) \tag{7.111}$$

choosing the negative sign for making the following equations more intuitive. With $\boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{y}^a) = \boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{l}) + Y^\mathsf{T}\boldsymbol{v}^a$ and $\boldsymbol{v} = \boldsymbol{v}^a + \Delta\boldsymbol{y}$ we have, up to first order, $\boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{y}^a) + Y^\mathsf{T}\Delta\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{l}) + Y^\mathsf{T}\boldsymbol{v}$ and therefore we can rewrite (7.110) as

$$\boldsymbol{c}_g - Y^\mathsf{T}\boldsymbol{v} = X \Delta\boldsymbol{x} \,. \tag{7.112}$$

[7] Observe, we do not have the estimated residuals $\boldsymbol{v}$ in the optimization function, but their corrections $\boldsymbol{v} - \boldsymbol{v}^a = \Delta\boldsymbol{y}$ (7.93), in order to be able to handle observations, such as directions or rotations, where a non-linear update of the observations is more appropriate, replacing (7.93), see e.g., (7.84).

[8] Again we do not indicate, that $\boldsymbol{c}_g$ depends on approximate values thus omit a superscript ^{a}.

[9] If we have the special linear Gauss–Markov model $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) = X\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{0}$, thus $Y = -I$, and use the approximate values $\boldsymbol{x}^a = \boldsymbol{0}$, then we have $\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{l}) = \boldsymbol{l}$.

Algorithm 2: Estimation in the Gauss–Helmert model.
$[\hat{\boldsymbol{x}}, \Sigma_{\hat{x}\hat{x}}, \hat{\sigma}_0^2, R]$ = `GaussHelmertModell_D`$(\boldsymbol{l}, \Sigma_{ll}, c_g, u_x, u_y, \boldsymbol{x}^a, \boldsymbol{\sigma}_{\hat{x}}^a, T_\theta,$ `maxiter`$)$
Input: observed values $\{\boldsymbol{l}, \Sigma_{ll}\}$, number N,
constraint function $[\boldsymbol{c}_g, X, Y] = c_g(\boldsymbol{l}, \boldsymbol{y}^a, \boldsymbol{x}^a)$, number G,
update functions u_x and u_y for the parameters and mean observations.
approximate values $\boldsymbol{x}^a$, possibly $\sigma_{\hat{\theta}_u}^a$,
parameters T_θ, `maxiter` for controlling convergence.
Output: estimated parameters $\{\hat{\boldsymbol{x}}, \Sigma_{\hat{x}\hat{x}}\}$, variance factor $\hat{\sigma}_0^2$, redundancy R.

Redundancy $R = G - U$;
if $R < 0$ **then stop**, not enough constraints;
Initiate: iteration $\nu = 0$, approximate values $\boldsymbol{y}^a := \boldsymbol{l}$, stopping variable: $s = 0$;
repeat
  Constraints and Jacobians : $[\boldsymbol{c}_g, X, Y] = c_g(\boldsymbol{l}, \boldsymbol{y}^a, \boldsymbol{x}^a)$, see (7.99), (7.96);
  Weight matrix of constraints: $W_{gg} = (Y^\mathsf{T}\Sigma_{ll}Y)^{-1}$;
  Build normal equation system: $[N, \boldsymbol{m}]$, see (7.100);
  if N is singular **then stop**: normal equation matrix is singular;
  Updates of parameter vector: $\Delta\boldsymbol{x}$, see (7.78), $\boldsymbol{x}^a := u_x(\boldsymbol{x}^a, \Delta\boldsymbol{x})$;
  Corrections for fitted observations: $\Delta\boldsymbol{y}$, see (7.103);
  Update fitted observations: $\boldsymbol{y}^a = u_y(\boldsymbol{y}^a, \Delta\boldsymbol{y})$, see (7.79);
  Set iteration: $\nu := \nu + 1$;
  if $\max_u(|\widehat{\Delta x}_u|/\sigma_{\hat{x}_u}^a) < T_x$ or $\nu =$ `maxiter` **then** $s = 2$;
until $s \equiv 2$;
Estimated parameters $\hat{\boldsymbol{x}} := \hat{\boldsymbol{x}}^a$ and covariance matrix: $\Sigma_{\hat{x}\hat{x}}$, see (7.106);
if $R > 0$ **then** variance factor $\hat{\sigma}_0^2 = \boldsymbol{c}_g^\mathsf{T} W_{gg}\, \boldsymbol{c}_g / R$;
else $\hat{\sigma}_0^2 = 1$;

Now, we define the substitution

$$\boldsymbol{v}_g = -Y^\mathsf{T}\boldsymbol{v}\,. \tag{7.113}$$

This is that part of the residuals $\boldsymbol{v}$ of the observations $\boldsymbol{l}$, which is relevant for the constraints. Its uncertainty results from (7.112), since $\Delta\boldsymbol{x}$ is assumed to be fixed in this step,

$$\mathbb{D}(\underline{\boldsymbol{c}}_g) = \mathbb{D}(-\boldsymbol{g}(\boldsymbol{x}^a, \underline{\boldsymbol{l}})) = \Sigma_{gg} = Y^\mathsf{T}\Sigma_{ll}Y\,. \tag{7.114}$$

We thus arrive at a representation of the functional model which has the algebraic structure of a Gauss-Markov model with $\boldsymbol{c}_g$ as observations and $\Delta\boldsymbol{x}$ as unknowns

quasi Gauss-Markov model representing the Gauss–Helmert model

$$\boxed{\boldsymbol{c}_g + \boldsymbol{v}_g = X\Delta\boldsymbol{x} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{c}}_g) = \Sigma_{gg}\,.} \tag{7.115}$$

Starting from here we solve the optimization problem for determining the corrections $\Delta\boldsymbol{x}$

$$\text{minimize} \qquad \boldsymbol{v}_g\, \Sigma_{c_g c_g}^{-1}\, \boldsymbol{v}_g \tag{7.116}$$
$$\text{subject to} \qquad -(\boldsymbol{c}_g + \boldsymbol{v}_g) + X\Delta\boldsymbol{x} = \boldsymbol{0}\,, \tag{7.117}$$

for given the contradictions $\boldsymbol{c}_g$ of the constraints, the Jacobian X, and the covariance matrix $\Sigma_{c_g c_g}$. As we know from the estimation with the Gauss–Markov model, we obtain the normal equation system

$$\boxed{N\Delta\boldsymbol{x} = \boldsymbol{m} \quad \text{with} \quad N = X^\mathsf{T}(Y^\mathsf{T}\Sigma_{ll}Y)^{-1}X \quad \text{and} \quad \boldsymbol{m} = X^\mathsf{T}(Y^\mathsf{T}\Sigma_{ll}Y)^{-1}\,\boldsymbol{c}_g\,.} \tag{7.118}$$

Hence the updated parameters are

$$\boldsymbol{x}^a := u_x(\boldsymbol{x}^a, \Delta\boldsymbol{x})\,. \tag{7.119}$$

We, however, need to be aware of the following: both, the coefficient matrix X and – via the Jacobian Y – the covariance matrix Σ_{gg} in (7.114) generally depend on the current values $\boldsymbol{x}^a$ and $\boldsymbol{y}^a$, since the Jacobians have to be determined at these values, see (7.96). So, we need to determine updates $\Delta\boldsymbol{y}$ for the mean observations $\boldsymbol{y}$ within the iterative scheme.

Since its Jacobian and covariance matrix depend on the unknown parameters we call thus functional the *quasi Gauss–Markov model* replacing the implicit constraints in the Gauss–Helmert model.[10]

7.5.2.2 Update of approximate fitted observations

From (7.115) and (7.113) we have the residuals $\boldsymbol{v}_g$ at some point within the iteration scheme

$$\boldsymbol{v}_g = -\boldsymbol{c}_g + X\Delta\boldsymbol{x} = -Y^\mathsf{T}\boldsymbol{v}\,, \tag{7.120}$$

which result after finding the locally best corrections $\Delta\boldsymbol{x}$. If we could determine the residuals $\boldsymbol{v}$ of the original observations from the residuals $\boldsymbol{v}_g$, i.e., invert the relation $\boldsymbol{v}_g = -Y^\mathsf{T}\boldsymbol{v}$, we could derive the corrections

$$\Delta\boldsymbol{y} = -\boldsymbol{v}^a + \boldsymbol{v}\,. \tag{7.121}$$

We could use them to determine updates for the mean observations $\boldsymbol{y}$. We obviously cannot determine the residuals $\boldsymbol{v}$ of the original observations by inversion of (7.113), since the matrix Y in generally does not have full rank.

Therefore we determine those residuals $\boldsymbol{v}$ which fulfil the constraint (7.120) and minimize $\Omega = \boldsymbol{v}^\mathsf{T}\Sigma_{ll}^{-1}\boldsymbol{v}$. With the Lagrangian parameter vector $\boldsymbol{\lambda}$ we thus need to

$$\boxed{\begin{array}{lll}\text{CONSTR}(\Sigma): & \text{minimize} & \boldsymbol{v}^\mathsf{T}\Sigma_{ll}^{-1}\boldsymbol{v} \\ & \text{subject to} & Y^\mathsf{T}\boldsymbol{v} + \boldsymbol{v}_g = \boldsymbol{0}\,.\end{array}} \tag{7.122}$$

w.r.t. the residuals $\boldsymbol{v}$. Setting the partials of

$$\boldsymbol{\Phi}(\boldsymbol{v},\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{v}^\mathsf{T}\Sigma_{ll}^{-1}\boldsymbol{v} + \boldsymbol{\lambda}^\mathsf{T}(Y^\mathsf{T}\boldsymbol{v} + \boldsymbol{v}_g) \tag{7.123}$$

to $\boldsymbol{0}$ yields the two necessary equations for $\boldsymbol{v}$

$$\frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{v}^\mathsf{T}} = \Sigma_{ll}^{-1}\boldsymbol{v} + Y\boldsymbol{\lambda} = \boldsymbol{0} \quad \text{and} \quad \frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{\lambda}^\mathsf{T}} = Y^\mathsf{T}\boldsymbol{v} + \boldsymbol{v}_g = \boldsymbol{0}\,. \tag{7.124}$$

From the first equation we obtain

$$\boldsymbol{v} = -\Sigma_{ll}Y\boldsymbol{\lambda} \tag{7.125}$$

which from the second equation leads to

$$\widehat{\boldsymbol{v}}_g - Y^\mathsf{T}\Sigma_{ll}Y\boldsymbol{\lambda} = \boldsymbol{0} \tag{7.126}$$

Therefore we have

$$\boldsymbol{\lambda} = (Y^\mathsf{T}\Sigma_{ll}Y)^{-1}\boldsymbol{v}_g\,, \tag{7.127}$$

which finally yields

$$\widehat{\boldsymbol{v}} = -\Sigma_{ll}Y(Y^\mathsf{T}\Sigma_{ll}Y)^{-1}\widehat{\boldsymbol{v}}_g \tag{7.128}$$

inverting the substitution in (7.113) in an intuitive manner. Hence, from (7.121) and (7.90) we obtain the corrections

$$\Delta\boldsymbol{y} = -u_y^{-1}(\boldsymbol{y}^a,\boldsymbol{l}) - \Sigma_{ll}Y(Y^\mathsf{T}\Sigma_{ll}Y)^{-1}\,\boldsymbol{g}(\boldsymbol{x}^a,\boldsymbol{l})\,. \tag{7.129}$$

[10] In the German geodetic literature on *adjustment theory* (equivalent to the estimation theory) the functional model (7.115) is called *'quasi vermittelnde Ausgleichung'*, derived from the German *'vermittelnde Ausgleichung'* representing the Gauss–Markov model. This motivates the English naming of this functional model, which only occurs as substitute for the linearized Gauss–Helmert model

If the residuals/corrections and the parameters/observations have the same dimension, e.g., for classical point coordinates, this simplifies to

$$\Delta \boldsymbol{y} = -\boldsymbol{y}^a + \boldsymbol{l} - \Sigma_{ll} Y (Y^\mathsf{T} \Sigma_{ll} Y)^{-1}\, \boldsymbol{g}(\boldsymbol{x}^a, \boldsymbol{l}))\,. \tag{7.130}$$

The update for the estimates of the mean observations then read as

$$\boldsymbol{y}^a := u_y(\boldsymbol{y}^a, \Delta \boldsymbol{y}) \quad \text{especially} \quad \widehat{\boldsymbol{y}}^a := \widehat{\boldsymbol{y}}^a + \Delta \boldsymbol{y}\,. \tag{7.131}$$

If the observations and the constraints, $\boldsymbol{l}$ and $\boldsymbol{g}$ are grouped as $\boldsymbol{l} = [\boldsymbol{l}_i]$ and $\boldsymbol{g} = [\boldsymbol{g}_i]$, such that each group $\boldsymbol{g}_i$ only refers to the corresponding group $\boldsymbol{l}_i$ and the observational groups are mutually independent, i.e., for $i \neq j$ we have $\mathbb{C}\text{ov}(\underline{\boldsymbol{l}}_i, \underline{\boldsymbol{l}}_j) = \boldsymbol{0}$, then with $\boldsymbol{v}_i^a$ the updates can be done group wise:

correction of estimated observations

$$\boxed{\Delta \boldsymbol{y}_i = u_y^{-1}(\boldsymbol{l}_i, \boldsymbol{y}_i^a) - \Sigma_{l_i l_i} Y_i (Y_i^\mathsf{T} \Sigma_{l_i l_i} Y_i)^{-1}\, \boldsymbol{g}(\widehat{\boldsymbol{x}}, \boldsymbol{l}_i)\,,} \tag{7.132}$$

with the individual updates

$$\widehat{\boldsymbol{y}}_i^a := u_y(\widehat{\boldsymbol{y}}_i^a, \Delta \boldsymbol{y}_i)\,. \tag{7.133}$$

If the observational noise is small, the Jacobian Y can be determined at the observations $\boldsymbol{l}$ instead of at the current value $\boldsymbol{y}$ of the mean observations. Hence the update step in (7.79) then would be omitted. The evaluation still can be based on the estimated variance factor, which can be based on $\boldsymbol{c}_g$ alone, and the covariance matrix of the estimated parameters.

7.5.2.3 Final estimates and evaluation

The final estimates are derived from the approximate values in the last iteration, assuming convergence is achieved. Hence we have the final estimates

$$\widehat{\boldsymbol{x}} := \boldsymbol{x}^a\,, \quad \widehat{\boldsymbol{y}} := \boldsymbol{y}^a\,, \quad \text{and} \quad \widehat{\boldsymbol{v}} = \boldsymbol{v}^a\,. \tag{7.134}$$

The estimated variance factor uses the value of the optimization function at the estimates and can be written in different ways

estimated variance factor

$$\widehat{\sigma}_0^2 = \frac{\Omega(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}})}{G-U} = \frac{\widehat{\boldsymbol{v}}^\mathsf{T} W_{ll} \widehat{\boldsymbol{v}}}{G-U} = \frac{\boldsymbol{c}_g^\mathsf{T} W_{gg} \boldsymbol{c}_g}{G-U} \tag{7.135}$$

The last relation can be derived at the point of convergence where $\Delta \boldsymbol{x} = \boldsymbol{0}$, $\Delta \boldsymbol{y} = \boldsymbol{0}$, and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$, using $\boldsymbol{c}_g = Y^\mathsf{T} \boldsymbol{v}$. Hence, the optimization can be based on the weighted sums of the squares of the estimated residuals $\boldsymbol{v}$ or the contradiction $\boldsymbol{c}_g$ of the constraints.

Finally, the theoretical covariance matrix of the estimated parameters can be derived from (7.118) by variance propagation, leading to the Cramer-Rao bound for the uncertainty

Cramer-Rao bound, covariance matrix of estimated parameters

$$\Sigma_{\widehat{x}\widehat{x}} = (X^\mathsf{T} (Y^\mathsf{T} \Sigma_{ll} Y)^{-1} X)^{-1}\,. \tag{7.136}$$

8 Variance Component Estimation with Observational Groups

Variance component estimation aims at statistically optimal finding factors for correcting additive components of the covariance matrices of observations. This note provides an efficient way to estimate these factors if the observations can be partitioned into mutually uncorrelated groups, which however show correlations within the groups. This is essential for handling coordinates, e.g. of image or GPS coordinates

Remark: We change notation in order to highlight the meaning of the individual variables: Variance components are variance factors, thus obtain an index 0. The factors within the iteration process are renamed to f_j^2, since they do not have any meaning as variance factors. ◇

8.1 Preface

This note (2023) provides explicit equations for a less specialized situation of variance component estimation than described in Förstner and Wrobel (2016, Sect. 4.2.4) for statistically independent observations. It handles the case were we have statistically independent groups of observations but allow for full correlations within a group. This situation occurs if coordinates of 2D or 3D points are observed, say in images or by GPS.

8.2 Summary

We derive simple equations for variance component estimation with I observational groups $\boldsymbol{y}_i$ having correlated individual observations for the case of the model

$$\mathbb{D}(\boldsymbol{y}_i) = \boldsymbol{\Sigma}_{y_i y_i} := (\sigma_{01}^2 + \sigma_{02}^2 s_i^2)\boldsymbol{\Sigma}_{ii}^0 \quad \text{with} \quad i = 1, ..., I\,. \tag{8.1}$$

The assumption is, that the covariance matrix of the keypoint coordinates mainly depend (1) on the covariance matrix for the i-th keypoint

$$\boldsymbol{\Sigma}_{ii}^0 = \frac{\sigma_n^2}{N_i} \underbrace{\left(G_{\sigma_i}(x,y) * (\nabla g_i(x,y) \nabla^\mathsf{T} g_i(x,y)\right)}_{T_{ii}}{}^{-1} \tag{8.2}$$

locally derived from the image information $g_i(x, y)$ around the keypoint by smoothing the squared gradient and (2) the scale s_i provided by the Lowe detector. The intensity-based prediction of the covariance matrix Σ_{ii}^0 mainly depends on the scaled inverse structure tensor T_{ii} derived from a Gaussian window $G_\sigma(x, y)$ with radius σ, the image noise variance σ_n^2 and the number $N_i = 12\sigma_i^2 + 1$ of the relevant pixels in the window, see Förstner et al. (2009, p. 5). The two variance factors σ_{01}^2 and σ_{02}^2 control the constant and scale dependent effect.

8.3 Basics

8.3.1 The iteration process

All variance factors are close to 1 and have unit [1]. They are estimated in an iterative scheme, where in each iteration the variance factors are multiplicatively corrected until convergence:

$$\sigma_{0j}^{2,(\nu+1)} = f_j^{2\,(\nu)}\, \sigma_{0j}^{2\,(\nu)}, \quad \text{with} \quad j = 1, ..., J \tag{8.3}$$

After convergence, i.e., if the factors $f_j^{2\,(\nu)}$ converge to 1, say in the μ-th iteration, we obtain the final estimates

$$\widehat{\sigma_{0j}^2} := \sigma_{0j}^{2\,(\mu)}. \tag{8.4}$$

8.3.2 The update factors f_j^2

We refer to Förstner and Wrobel (2016, Sect. 4.2.4) and start from (4.97) for the factors[1]

$$f_j^2 = \frac{\hat{\boldsymbol{v}}^\mathsf{T} \boldsymbol{W}_{yy} \boldsymbol{\Sigma}_j^a \boldsymbol{W}_{yy} \hat{\boldsymbol{v}}}{\mathrm{tr}(\boldsymbol{W}_{yy} \boldsymbol{\Sigma}_j^a \boldsymbol{W}_{yy} \boldsymbol{\Sigma}_{\hat{v}\hat{v}})} \quad \text{with} \quad \boldsymbol{W}_{yy} = \boldsymbol{\Sigma}_{yy}^{-1} \tag{8.5}$$

We have omitted the iteration index for simplicity.

In our case we have two variance components σ_{0j}^2, which are used in the following model

$$\boldsymbol{\Sigma}_{yy} = \sum_{j=1}^{2} \boldsymbol{\Sigma}_j = \sum_{j=1}^{2} \sigma_{0j}^2 \boldsymbol{\Sigma}_j^a \tag{8.6}$$

with the block diagonal matrices

$$\boldsymbol{\Sigma}_1^a = \mathrm{Diag}(\{\boldsymbol{\Sigma}_{1,ii}^a\}) = \mathrm{Diag}(\{\boldsymbol{\Sigma}_{ii}^0\}) \quad \text{and} \quad \boldsymbol{\Sigma}_2^a = \mathrm{Diag}(\{\boldsymbol{\Sigma}_{2,ii}^a\}) = \mathrm{Diag}(\{s_i^2 \boldsymbol{\Sigma}_{ii}^0\}), \tag{8.7}$$

replacing (4.93). Observe, we use the indices $i = 1, ..., I$ for the observational groups and replaced the unit matrix by the block diagonal matrix of the $\boldsymbol{\Sigma}_{ii}^0$.

8.4 Simplification

We first determine the weight matrix from the previous iteration

$$\boldsymbol{W} = \mathrm{Diag}(\{\boldsymbol{W}_{y_i y_i}\}) \tag{8.8}$$

with

$$\boldsymbol{W}_{y_i y_i} = w_i \boldsymbol{W}_{ii}^0, , \quad w_i = \frac{1}{\sigma_{01}^2 + \sigma_{02}^2 s_i^2} \quad \text{and} \quad \boldsymbol{W}_{ii}^0 = (\boldsymbol{\Sigma}_{ii}^0)^{-1}. \tag{8.9}$$

Thus, in the following derivation the variance components σ_{01}^2 and σ_{02}^2 are those from the previous iteration and taken as fixed values.

The variance components $(\sigma_{01}^2)^{(0)}$ and $(\sigma_{02}^2)^{(0)}$ may be chosen to be 1 at the beginning.

[1] denoted with f_i in order to clarify they are not the variance components and no estimates, just intermediate values within the iteration process

8.4.1 The first variance component

We obtain the (correcting) factor for the first variance component

$$f_1^2 = \frac{\hat{\boldsymbol{v}}^\mathsf{T} W_{yy}\Sigma_1^a W_{yy}\hat{\boldsymbol{v}}}{\mathrm{tr}(W_{yy}\Sigma_1^a W_{yy}\Sigma_{\hat{v}\hat{v}})} \tag{8.10}$$

$$= \frac{\hat{\boldsymbol{v}}^\mathsf{T}\mathrm{Diag}(\{W_{y_iy_i}\Sigma_{ii}^a W_{y_iy_i}\})\hat{\boldsymbol{v}}}{\mathrm{tr}\,(\mathrm{Diag}(\{W_{y_iy_i}\Sigma_{ii}^a W_{y_iy_i}\})\Sigma_{\hat{v}\hat{v}})} \tag{8.11}$$

Since the covariance and the weight matrix of the observations are block diagonal, and the individual covariance matrices have the common factor Σ_{ii}^0, the numerator can directly be written as a sum:

$$\hat{\boldsymbol{v}}^\mathsf{T}\mathrm{Diag}(\{W_{y_iy_i}\Sigma_{1,ii}^a W_{y_iy_i}\})\hat{\boldsymbol{v}} = \sum_{i=1}^{I}\hat{\boldsymbol{v}}_i^\mathsf{T} W_{y_iy_i}\Sigma_{ii}^a W_{y_iy_i}\boldsymbol{v}_i \tag{8.12}$$

$$= \sum_{i=1}^{I}\sigma_{01}^2\, w_i^2\, \hat{\boldsymbol{v}}_i^\mathsf{T} W_{ii}^0\hat{\boldsymbol{v}}_i \tag{8.13}$$

The denominator needs a bit care, since the covariance matrix $\Sigma_{\hat{v}\hat{v}}$ is a full matrix. However, the trace only requires the diagonal blocks, hence we have

$$\mathrm{tr}\,(\mathrm{Diag}(\{W_{y_iy_i}\Sigma_{ii}^a W_{y_iy_i}\})\Sigma_{\hat{v}\hat{v}}) = \sum_{i=1}^{I}\mathrm{tr}(W_{y_iy_i}\Sigma_{ii}^a W_{y_iy_i}\Sigma_{\hat{v}_i\hat{v}_i}) \tag{8.14}$$

$$= \sum_{i=1}^{I}\sigma_{01}^2\, w_i\,\mathrm{tr}(W_{y_iy_i}\Sigma_{\hat{v}_i\hat{v}_i}) \tag{8.15}$$

With the redundancy matrix

$$R = \Sigma_{\hat{v}\hat{v}}W_{yy} = [R_{i,i'}] \quad \text{with} \quad \mathrm{tr}R = \sum_{i=1}^{I}\mathrm{tr}R_{ii} = N - U \tag{8.16}$$

we have the redundancy contribution of each observational group

$$r_i := \mathrm{tr}R_{ii} \quad \text{with} \quad \sum_{i=1}^{I} r_i = R\,. \tag{8.17}$$

This is useful, since we might be able to approximate the local redundancy contributions.

Thus the factor for the first variance component is given by

$$\boxed{f_1^2 = \frac{\sum_{i=1}^{I}\sigma_{01}^2\, w_i^2\, \hat{\boldsymbol{v}}_i^\mathsf{T} W_{ii}^0\hat{\boldsymbol{v}}_i}{\sum_{i=1}^{I}\sigma_{01}^2\, w_i\, r_i}} \tag{8.18}$$

8.4.2 The second variance component

We similarly obtain the factor for the second component

$$f_2^2 = \frac{\hat{\boldsymbol{v}}^\mathsf{T} W_{yy}\Sigma_2^a W_{yy}\hat{\boldsymbol{v}}}{\mathrm{tr}(W_{yy}\Sigma_2^a W_{yy}\Sigma_{\hat{v}\hat{v}})} \tag{8.19}$$

$$= \frac{\hat{\boldsymbol{v}}^\mathsf{T}\mathrm{Diag}(\{W_{y_iy_i}\, s_i^2\Sigma_{ii}^a\, W_{y_iy_i}\})\hat{\boldsymbol{v}}}{\mathrm{tr}\,(\mathrm{Diag}(\{W_{y_iy_i}\, s_i^2\Sigma_{ii}^a\, W_{y_iy_i}\})\Sigma_{\hat{v}\hat{v}})} \tag{8.20}$$

Following similar arguments, we just need to weight the numerator and the denominator with a factor s_i^2 and obtain

$$\boxed{f_2^2 = \frac{\sum_{i=1}^{I} s_i^2\,\sigma_{01}^2\, w_i^2\, \hat{\boldsymbol{v}}_i^\mathsf{T} W_{ii}^0\hat{\boldsymbol{v}}_i}{\sum_{i=1}^{I} s_i^2\,\sigma_{01}^2\, w_i\, r_i}} \tag{8.21}$$

Remark: As a check we determine the sum of the two denominators:

$$\sum_{i=1}^{I} f_1^2\, w_i\, r_i + \sum_{i=1}^{I} s_i^2\, f_2^2\, w_i\, r_i = \sum_{i=1}^{I} (f_1^2 + s_i^2\, f_2^2)\, w_i\, r_i = \sum_{i=1}^{I} r_i = R\,, \tag{8.22}$$

which demonstrates that the redundancy R is distributed on the points and their two uncertainty components.

8.5 Approximating the redundancy contributions $\overline{r_i}$ for image points within bundle adjustment

If a scene point is observed from K images, then – assuming the poses are fixed – the triangulation uses $N = 2K$ observed image coordinates for the $U = 3$ scene coordinates, hence the redundancy of the triangulation is

$$R = 2K - 3\,. \tag{8.23}$$

Hence the redundancy number r_n for each coordinate on an average is $R/(2K)$. Therefore, the redundancy contribution of the points with two coordinates on an average is double this value, hence

$$\overline{r_i} = \frac{R}{K} = \frac{2K-3}{K}\,, \tag{8.24}$$

which we may use as an approximation for $\mathrm{tr}\mathsf{R}_{ii}$ during variance component estimation.

9 Pre-calibration and in-situ Self-calibration with Correlated Observations

Deformation analysis based on point clouds taken at different times may require to take into account both pre-calibration and in situ self-calibration of the used instruments. We analyse the mutual effect of pre-calibration and in-situ self-calibration w.r.t. (1) the necessity to exploit the full covariance structure of the point cloud induced by the pre-alibration and (2) the possibility of increasing the computational efficiency during the self-calibration.

9.1 Preface

The note (2023) addresses the question how a priori pre-calibration result may influence a possible in-situ self-calibration, both concerning the achievable accuracy as well as the numerical effort. The result uses a lemma by Rao (1967, Lemma 5a) which states under which conditions the result of an estimation is invariant to a change in the assumed structure of the covariance matrix of the observations.

9.2 Summary

We analyse the computational and statistical efficiency of self-calibration when reconstructing a surface from point cloud taken with a laser scanner where we know the calibration result. We discuss fusing the prior calibration information with the one from the in-situ measurements and the effect of the uncertainty of the prior calibration $(\boldsymbol{c}_a, \Sigma_{c_a c_a})$ onto the covariance matrix $\mathbb{D}(\underline{\boldsymbol{\theta}})$ of the unknown parameters $\boldsymbol{\theta}$.

We address four cases, A to D, differing by their stochastical and their functional models:

- The a priori calibration result is not available (or used, cases (A,C)) or is integrated into the self-calibration with a priori information in a Bayesian manner (cases (B,D)). Hence, we have the two (linearized) functional models for estimating the parameters $\boldsymbol{y}$ of the object's form and the calibration parameters $\boldsymbol{c}$

$$(A,C):\ \mathbb{E}\left(\underline{\boldsymbol{y}}\right) = [\mathsf{B}\,,\,\mathsf{C}]\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{c}\end{bmatrix},\ \text{or}\ (B,D):\ \mathbb{E}\left(\begin{bmatrix}\underline{\boldsymbol{y}}\\ \underline{\boldsymbol{c}}_a\end{bmatrix}\right) = \begin{bmatrix}\mathsf{B} & \mathsf{C}\\ \mathsf{0} & \mathsf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{c}\end{bmatrix}. \quad (9.1)$$

- The covariance matrix of the observations is assumed to be (a) block diagonal, assuming the **p**oints are mutually uncorrelated or to be (b) fully populated due to the joint effect of the uncertainty $\Sigma_{c_a c_a}$ of the a priori **c**alibration parameters $\boldsymbol{c}_a$ onto the observations $\mathsf{C}_a\boldsymbol{c}_a$. So we either use

$$\begin{aligned}(A):\ \mathbb{D}(\underline{\boldsymbol{y}}) &= \Sigma_{ll,p} =: \Sigma_0\,, \quad \text{or} \quad (C):\ \mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma_0 + \mathsf{C}_a\Sigma_{c_a c_a}\mathsf{C}_a^\mathsf{T} =: \Sigma\,. \quad (9.2)\\ (B):\ \mathbb{D}\left(\begin{bmatrix}\underline{\boldsymbol{y}}\\ \underline{\boldsymbol{c}}_a\end{bmatrix}\right) &= \begin{bmatrix}\Sigma_0 & \mathsf{0}\\ \mathsf{0} & \Sigma_{c_a c_a}\end{bmatrix} \quad \text{or} \quad (D):\ \mathbb{D}\left(\begin{bmatrix}\underline{\boldsymbol{y}}\\ \underline{\boldsymbol{c}}_a\end{bmatrix}\right) = \begin{bmatrix}\Sigma_0 & \mathsf{0}\\ \mathsf{0} & \mathsf{0}\end{bmatrix} + \begin{bmatrix}\mathsf{C}_a\\ \mathsf{I}\end{bmatrix}\Sigma_{c_a c_a}[\mathsf{C}_a^\mathsf{T}\,,\,\mathsf{I}]\,.\end{aligned}$$

The four cases are analysed w.r.t. their estimates and covariance matrices, see Tab. 9.1.

	$\mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma_0$		$\mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma$
SC	$\hat{\boldsymbol{\theta}} \mid A$ $\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) =$ $\begin{bmatrix}\mathsf{B}^\mathsf{T}\mathsf{W}_0\mathsf{B} & \mathsf{B}^\mathsf{T}\mathsf{W}_0\mathsf{C}\\ \mathsf{C}^\mathsf{T}\mathsf{W}_0\mathsf{B} & \mathsf{C}^\mathsf{T}\mathsf{W}_0\mathsf{C}\end{bmatrix}^{-1}$	$\equiv$	$\hat{\boldsymbol{\theta}} \mid C$ $\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) =$ $\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) + \begin{bmatrix}\mathsf{0}\\ \mathsf{I}_C\end{bmatrix}\Sigma_{c_a c_a}[\mathsf{0}\,,\,\mathsf{I}_C]$
BSC	$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid B) =$ $\begin{bmatrix}\mathsf{B}^\mathsf{T}\mathsf{W}_0\mathsf{B} & \mathsf{B}^\mathsf{T}\mathsf{W}_0\mathsf{C}\\ \mathsf{C}^\mathsf{T}\mathsf{W}_0\mathsf{B} & \mathsf{C}^\mathsf{T}\mathsf{W}_0\mathsf{C} + \mathsf{W}_{c_a c_a}\end{bmatrix}^{-1}$		$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid D) =$ D: $\begin{bmatrix}(\mathsf{B}^\mathsf{T}\mathsf{W}_0\mathsf{B})^{-1} & \mathsf{0}\\ \mathsf{0} & \Sigma_{c_a c_a}\end{bmatrix}$

Table 9.1: Estimates and covariance matrices of the estimated parameters when using the four models for self-calibration and assuming $\mathsf{C} = \mathsf{C}_a$. SC: self-calibration without prior, BSC: Bayesian self-calibration

The main result of this note is the following: *If the matrix $\mathsf{C} \equiv \mathsf{C}_a$ is common to the stochastical model in (9.2) and the functional models in (9.1), then, following Rao (1967, Lemma 5a), the estimates of model A and C coincide*, allowing to use model A for an efficient estimation of the parameters and their covariance matrix. Moreover, using Rao's lemma decorrelates and simplifies the solution for model D.

Especially, we have the following relations between the estimates in the model A and in the models B and C

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid B) \le \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) \le \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C)\,, \quad (9.3)$$

$$\mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid D) \le \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid B) \le \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid A) = \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid C)\,. \quad (9.4)$$

An individual sensitivity analysis allows to determine the expected loss in quality, accuracy and reliability in Baarda's sense, without requiring actual observations.

9.3 Introduction

9.3.1 Motivation

Taking point clouds as observations for the estimation of object forms, for deformation analysis, or for calibration needs to take the stochastical properties of the coordinates of the points into account as far as necessary. The quality of the assumed stochastical model needs to be *acceptable*, not necessarily optimal, *for the envisaged application*.

Especially for deformation analysis, where the deformations are in the order of the measuring precision, a realistic stochastical model, taking all known dependencies into account, may be required.

Unfortunately, the points in a point cloud may be correlated due to the uncertainty of the instrumental calibration. This generally leads to a large fully populated covariance matrix Σ_{yy} of the N observations, collected in the vector $\boldsymbol{y}$. As a consequence any estimation minimizing the weighed squares of the residuals is confronted with using the inverse $W_{yy} = \Sigma_{yy}$, which often is called information matrix or precision matrix.

This note shows under which conditions it is possible to work with uncorrelated points, thus with a block matrix containing the 3×3 covariance matrices $\Sigma_{y_i y_i}$ of the I individual points, instead of a fully populated covariance matrix, without losing accuracy.

9.3.2 Rao's lemma

The idea is to exploit the Lemma 5a in Rao (1967) which states under which conditions the estimation with a covariance matrix containing certain additive variance components does not change the parameters. Especially, it starts from the given the linear Gauss-Markov model,

$$\boldsymbol{y} + \boldsymbol{v} = \boldsymbol{X}\boldsymbol{\theta} \quad \text{with} \quad \Sigma_0 = \mathbb{D}(\boldsymbol{y})\,, \tag{9.5}$$

and the estimated parameters

$$\hat{\boldsymbol{\theta}}_0 = (\boldsymbol{X}^\mathsf{T}\Sigma_0^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\Sigma_0^{-1}\boldsymbol{y}\,, \tag{9.6}$$

Then, when using the modified covariance matrix

$$\Sigma = \boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^\mathsf{T} + \Sigma_{\mathbf{0}}\boldsymbol{Z}\boldsymbol{\Theta}\boldsymbol{Z}^\mathsf{T}\Sigma_{\mathbf{0}} + \Sigma_{\mathbf{0}} \quad \text{with} \quad \boldsymbol{Z}^\mathsf{T}\boldsymbol{X} = \boldsymbol{0}\,, \tag{9.7}$$

with arbitrary matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Theta}$ (which we will not need in the following) the estimate $\hat{\boldsymbol{\theta}}_0$ from (9.6) is identical to the estimate,

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\mathsf{T}\Sigma^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\Sigma^{-1}\boldsymbol{y}\,, \tag{9.8}$$

when using the full covariance matrix.

Fig. 9.1 shows the principle of least squares estimation with a unit matrix and an arbitrary covariance matrix for the observations in the simple model $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{x}\theta, \Sigma)$. Fig. 9.2 visualizes the idea of Rao's lemma.

As can be seen by variance propagation its covariance matrix is

$$\Sigma_{\hat{\theta}\hat{\theta}} = (\boldsymbol{X}^\mathsf{T}\Sigma^{-1}\boldsymbol{X})^{-1}\,, \tag{9.9}$$

hence, not $(\boldsymbol{X}^\mathsf{T}\Sigma_0^{-1}\boldsymbol{X})^{-1}$, thus in principal needs to take the full covariance matrix Σ into account.[1]

Observe, the two first components in the covariance matrix (9.7) have the structure of a weighted block dyadic product $\boldsymbol{X}\boldsymbol{S}\boldsymbol{X}^\mathsf{T}$, similar to the 1D case $s\,\boldsymbol{x}\boldsymbol{x}^\mathsf{T}$.

[1] The result has as special case the mean of N values y_n in case the observations have the same variance σ^2 and are mutually correlated with the same correlation coefficient $\rho \in [-1/(N-1), +1]$. Then the normal arithmetic mean $\hat{\mu} = \sum_n l_n/N$ is the optimal estimator, but its variance is $\sigma_{\hat{\mu}}^2 = (1 + r(N-1)) \cdot \sigma^2/N$, but not σ^2/N. This can be shown using $\boldsymbol{A} = \mathbf{1}_N$, thus a vector of N ones, and $\Sigma = \sigma^2[(1-\rho)I_N + \rho\mathbf{1}_N\mathbf{1}_N^\mathsf{T}]$.

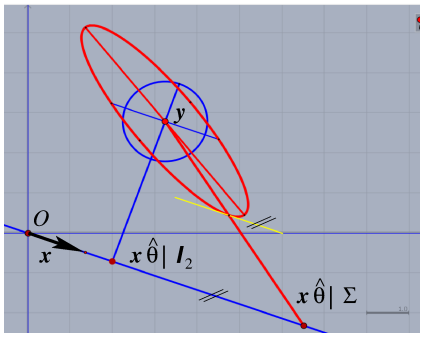


Figure 9.1: Least squares estimation. Model $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{x}\theta, \Sigma)$. If $\Sigma = I_2$, indicated by the blue circular standard ellipse, the optimal point lies on the footpoint $\hat{\boldsymbol{y}} \mid I_2 = \boldsymbol{x}\hat{\theta} \mid I_2$ of $\boldsymbol{y}$ onto the line $\mathbb{E}(\underline{\boldsymbol{y}}) = \boldsymbol{x}\theta$. If the covariance matrix Σ is a general matrix, represented by the red standard ellipse, then the optimal point $\hat{\boldsymbol{y}} \mid \Sigma = \boldsymbol{x}\hat{\theta} \mid \Sigma$ is the intersection of the (blue) line $\mathbb{E}(\underline{\boldsymbol{y}}) = \boldsymbol{a}\theta$ passing through O and the (red) line, defined by the direction from $\boldsymbol{y}$ to that point of the ellipse, where the tangent (yellow) is parallel to $\boldsymbol{x}$

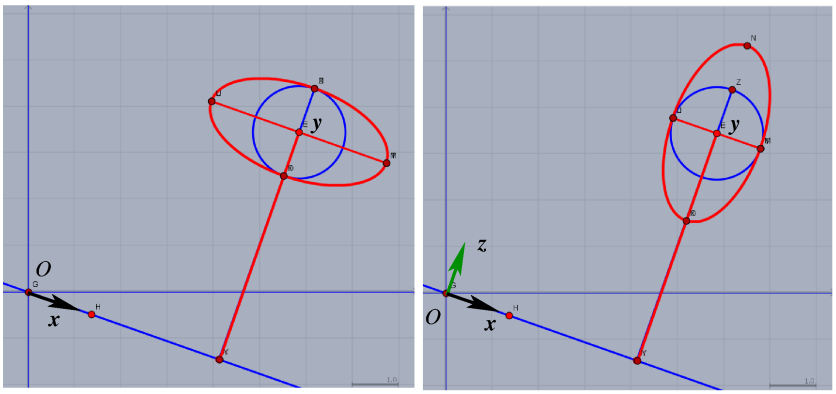


Figure 9.2: Visualization of Rao's lemma: Least squares estimation with modified covariance matrix. Model $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{x}\theta, \Sigma)$. If $\Sigma = \sigma^2 I + \gamma \boldsymbol{x}\boldsymbol{x}^\mathsf{T}$ or if $\Sigma = \sigma^2 I + \theta \boldsymbol{z}\boldsymbol{z}^\mathsf{T}$, with $\boldsymbol{z} \perp \boldsymbol{x}$, hence generally, if $\Sigma = \sigma^2 I + \gamma \boldsymbol{x}\boldsymbol{x}^\mathsf{T} + \theta \boldsymbol{z}\boldsymbol{z}^\mathsf{T}$, the semi-axes of the standard ellipse are parallel or orthogonal to $\boldsymbol{x}$. Then, the least squares estimate for the generalized covariance matrix is the same as for $\Sigma = I_2$. However, the covariance matrix of the estimate depends on the modification, namely the factors σ^2 and γ.

Remark: It is well known[2], that changing the covariance matrix of the observations leads to an effect onto the estimated parameter, which is in the range of their standard deviations, unless the change of the covariance matrix or the weight matrix is very large. It mainly influences their covariance matrix. The result of Rao's Lemma addresses the extreme case, where the effect onto the parameters is zero, which requires that the change of the covariance matrix has a special structure. The effect of model errors has been discussed in the context of self-calibration in Förstner (1982). ◇

9.3.3 Goal and result

The idea is to choose the matrices X, Σ_0, $\mathbf{\Gamma}$, and $\mathbf{\Theta}$ in (9.7), such, that the estimation of the parameters for the object and the calibration

1. can be performed within self-calibration with a block diagonal matrix for the observed points, which increases computational efficiency
2. can use the parameters of a priori calibration for an in-situ self-calibration possibly improving these parameters, and
3. efficiently derive the uncertainty of the estimated parameters.

Computational efficiency also can be achieved, in case only a part of the calibration parameters is included in the self-calibration. The increase in efficiency refers to the estimation of the parameters, as well as to determination of their covariance matrix.

9.4 The setup

We now discuss the used stochastical model of the observations and then four mathematical models for the self-calibration

9.4.1 The covariance matrix of the observations

We assume two sources of measurement deviations, (1) caused by the object properties, yielding $\Sigma_{yy,p}$, and (2) caused by the prior calibration, yielding $\Sigma_{yy,c}$. Hence, we assume the complete uncertainty is described by

$$\Sigma_{yy} = \Sigma_{yy,p} + \Sigma_{yy,c}\,. \tag{9.10}$$

1. The covariance matrix $\Sigma_{yy,p}$ is assumed to be block diagonal

   $$\Sigma_{yy,p} = \mathrm{Diag}(\{\Sigma_{y_iy_i,p}\})\,. \tag{9.11}$$

   and has full rank. The individual 3×3 covariance matrices $\Sigma_{y_iy_i,p}$ are assumed to reflect those parts of the directional and distance uncertainties, which are independent for each point, including those parts which depend on the surface point, e.g., its material and the impact angle.

2. The covariance matrix $\Sigma_{yy,c}$ is assumed to contain all uncertainties of the a priori calibration which effect all points of a scan simultaneously. We do not assume other types of correlations, e.g., caused by the atmosphere. Using the primary error concept the effect of the C calibration parameters[3] $\boldsymbol{c}$ onto the observations is assumed to be describable by

   $$\boldsymbol{y}_c = C_a \boldsymbol{c}_a\,. \tag{9.12}$$

[2] See Koch (1999), where eq. (3.108) shows the effect of using a slightly changed weight matrix $W+\Delta W$ instead of W, and with (3.32) reads as $\hat{\boldsymbol{\theta}} \mid (W+\Delta W) \approx \hat{\boldsymbol{\theta}} \mid W - (X^\mathsf{T} W X)^{-1} X^\mathsf{T} \Delta W\, \hat{\boldsymbol{e}}$, with the estimated residuals $\hat{\boldsymbol{e}} = \boldsymbol{y} - X\hat{\boldsymbol{\theta}}$.

[3] We assume a perfectly constructed instrument would lead to $\boldsymbol{c} = 0$.

Assuming a linear model appears to be reasonable as the effects are small. The estimated parameters $\boldsymbol{c}_a := \widehat{\boldsymbol{c}}_a$ of the a priori calibration will be uncertain[4]

$$\underline{\boldsymbol{c}}_a \sim \mathcal{N}(\boldsymbol{\mu}_{c_a}, \Sigma_{c_a c_a})\,. \tag{9.13}$$

This leads to the uncertain effects of the calibration onto the observations

$$\underline{\boldsymbol{y}}_c \sim \mathcal{N}(\boldsymbol{\mu}_{y_c}, \Sigma_{yy,c_a}) \quad \text{with} \quad \boldsymbol{\mu}_{y_c} = \mathsf{C}_a \boldsymbol{\mu}_{c_a} \quad \text{and} \quad \Sigma_{yy,c_a} = \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}}\,. \tag{9.14}$$

The covariance matrix has a low rank $C = \mathrm{rk}(\mathsf{C}_a)$, but generally is fully populated.

Hence, also the covariance matrix Σ_{yy} will be fully populated since it has the structure

$$\Sigma_{yy} = \Sigma_{yy,p} + \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}}\,, \tag{9.15}$$

where the first part is sparse, namely block-diagonal, and the second part has the structure of a block dyadic product.

On notation: In the following we denote the inverses of covariance matrices as weight/precision/- or information matrices:

$$\mathsf{W} = \Sigma^{-1}\,, \quad \mathsf{W}_0 = \Sigma_0^{-1}\,, \quad \mathsf{W}_{yy,p} = \Sigma_{yy,p}^{-1}\,, \quad \text{and} \quad \mathsf{W}_{c_0 c_0} = \Sigma_{c_0 c_0}^{-1}\,. \tag{9.16}$$

We now discuss four cases for the self-calibration, which simultaneously determines the parameters of the object and calibration parameters. We assume two alternatives for the functional model of the self-calibration and two alternatives for the stochastical model for the observations. Hence, we arrive at the following models

A. Uncorrelated points for self-calibration

B. Uncorrelated points for self-calibration with fusion of the prior calibration

C. Correlated points for self-calibration

D. Correlated points for self-calibration with fusion of the prior calibration

1. The a priori calibration result (A,C) is not available or used or (B,D) is fusing the self-calibration with a priori information. Hence, we have the two (linearized) functional models for estimating the parameters $\boldsymbol{y}$ of the object's form and the calibration parameters $\boldsymbol{c}$

$$(A,C):\ \mathbb{E}\left(\underline{\boldsymbol{y}}\right) = [\mathsf{B}\,,\mathsf{C}] \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix}\,, \quad \text{or} \quad (B,D):\ \mathbb{E}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix}\right) = \begin{bmatrix} \mathsf{B} & \mathsf{C} \\ \mathsf{0} & \mathsf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix}\,. \tag{9.17}$$

We call models (A,C) *self-calibration* and models (B,D) *self-calibration with fusion* in the following.
Depending on the context, the self-calibration may refer to only a subset of parameters used in the prior calibration, e.g., only those which are to be expected to be determinable within the self-calibration. Similarly, self-calibration with fusion (2) may only refer to those parameters which are expected to change over time. Hence, the models may have the same coefficient matrix B but different coefficient matrices C.

2. The covariance matrix of the observations is assumed to be block diagonal, assuming the **p**oints are mutually uncorrelated or to be fully populated due to the joint effect of the uncertainty $\Sigma_{c_a c_a}$ of the a priori **c**alibration parameters $\boldsymbol{c}_a$ onto the observations $\mathsf{C}_a \boldsymbol{c}_a$.

[4] Random variables are underscored.

Since the observation vectors in models (A,C) are different from thos in models (B,D) we need to consider them separately.
In case of models (A,C) we either use

$$\mathbb{D}(\boldsymbol{y} \mid A) = \Sigma_{yy,p} =: \Sigma_0\,, \tag{9.18}$$

or

$$\mathbb{D}(\boldsymbol{y} \mid C) = \Sigma_{yy,p} + \Sigma_{yy,c} = \Sigma_{yy,p} + \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} =: \Sigma\,. \tag{9.19}$$

Stochastical model (A) is a special case of model (C), so, when used, leads to sub-optimal estimates, if the observations actually are correlated.
In case of models B the a priori information ($\boldsymbol{c}_a, \Sigma_{c_a c_a}$ used in the self-calibration with fusion we reasonably may assume the prior information is independent of the observed points, hence we have

$$\mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \mid B\right) = \begin{bmatrix} \Sigma_{yy,p} & \mathsf{0} \\ \mathsf{0} & \Sigma_{c_a c_a} \end{bmatrix}. \tag{9.20}$$

In case D we assume the observed coordinates are mutually correlated due to the common calibration uncertainty. But, then also the calibration parameters $\boldsymbol{c}_a$ will be correlated with the observed points, since we have

$$\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{y}}_g + \mathsf{C}_a \underline{\boldsymbol{c}}_a \\ \underline{\boldsymbol{c}}_a \end{bmatrix} = \begin{bmatrix} \mathsf{I} & \mathsf{C}_a \\ \mathsf{0} & \mathsf{I} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{\theta}} \\ \underline{\boldsymbol{c}} \end{bmatrix}. \tag{9.21}$$

Hence we obtain the joint covariance matrix

$$\begin{aligned} \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \mid D\right) &= \begin{bmatrix} \mathsf{I} & \mathsf{C} \\ \mathsf{0} & \mathsf{I} \end{bmatrix} \begin{bmatrix} \Sigma_{yy,p} & \\ & \Sigma_{c_a c_a} \end{bmatrix} \begin{bmatrix} \mathsf{I} & \mathsf{0} \\ \mathsf{C}^{\mathsf{T}} & \mathsf{I} \end{bmatrix} && (9.22) \\ &= \begin{bmatrix} \Sigma_{yy,p} + \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \mathsf{C}_a \Sigma_{c_a c_a} \\ \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \Sigma_{c_a c_a} \end{bmatrix} && (9.23) \\ &= \begin{bmatrix} \Sigma_{yy,p} & \mathsf{0} \\ \mathsf{0} & \mathsf{0} \end{bmatrix} + \begin{bmatrix} \mathsf{C}_a \\ \mathsf{I} \end{bmatrix} \Sigma_{c_a c_a} [\mathsf{C}_a^{\mathsf{T}}, \mathsf{I}] && (9.24) \end{aligned}$$

There is a profound difference when fusing the uncorrelated and the correlated observations in models B and D.
We first look at the models A and C. As can be seen from (9.18) and (9.19), the uncertainty does not decrease when taking the correlations into account:

$$\mathbb{D}(\boldsymbol{y} \mid C) - \mathbb{D}(\boldsymbol{y} \mid A) = \Sigma - \Sigma_0 = \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} \geq 0 \tag{9.25}$$

Hence, the uncertainty in model C generally is higher than in model A. Hence, we can expect, the results using model B are worse (not better) than those with model A. Since the two groups of observations is not independent the model does represent a Bayesian estimation of the parameters.
This contrasts to the relation between the uncertainties in models B and D. Here we have

$$\begin{aligned} \mathbb{D}(\boldsymbol{y} \mid D) - \mathbb{D}(\boldsymbol{y} \mid B) &= \begin{bmatrix} \Sigma_{yy,p} + \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \mathsf{C}_a \Sigma_{c_a c_a} \\ \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \Sigma_{c_a c_a} \end{bmatrix} - \begin{bmatrix} \Sigma_{yy,p} & \mathsf{0} \\ \mathsf{0} & \Sigma_{c_a c_a} \end{bmatrix} \\ &= \begin{bmatrix} \mathsf{C}_a \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \mathsf{C}_a \Sigma_{c_a c_a} \\ \Sigma_{c_a c_a} \mathsf{C}_a^{\mathsf{T}} & \mathsf{0} \end{bmatrix} \gtreqless 0\,. \end{aligned} \tag{9.26}$$

Hence, the accuracy difference is indefinite. This indicates, that model D will not generally lead to better results than model B. Also, since the two groups of observations is not independent the model does not represent a Bayesian estimation of the parameters.

We now discuss the four different models in more detail.

9.4.2 A: Self-calibration with independent points

The most simple model, case A, is in-situ self-calibration *without* having access (or using) to the result of a prior calibration. It reads as

$$\mathbb{E}(\underline{\boldsymbol{y}} \mid A) = \boldsymbol{X}\boldsymbol{\theta} \quad \mathbb{D}(\underline{\boldsymbol{y}} \mid A) = \boldsymbol{\Sigma}_0 \quad \text{with} \quad \boldsymbol{X} = [\boldsymbol{B}, \boldsymbol{C}], \quad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix} \tag{9.27}$$

where

- the Y parameters $\boldsymbol{y}$ are used to describe the object, e.g., using splines, and
- the C parameters $\boldsymbol{c}$ are calibration parameters within the self-calibrating estimation. They generally need not be the same as in a pre-calibration.
- Since we do not use or have access to a prior calibration, we need to assume the covariance matrix of the observations is block diagonal.

The uncertainty of the estimated parameters results from the normal equations

$$\boldsymbol{N}_0\hat{\boldsymbol{\theta}} = \boldsymbol{n}_0 \tag{9.28}$$

with

$$\boldsymbol{N}_0 = \begin{bmatrix} \boldsymbol{N}_{11,0} & \boldsymbol{N}_{12,0} \\ \boldsymbol{N}_{21,0} & \boldsymbol{N}_{22,0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{B} & \boldsymbol{B}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{C} \\ \boldsymbol{C}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{B} & \boldsymbol{C}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{C} \end{bmatrix} \tag{9.29}$$

and

$$\boldsymbol{n}_0 = \begin{bmatrix} \boldsymbol{n}_{1,0} \\ \boldsymbol{n}_{2,0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{y} \\ \boldsymbol{C}^\mathsf{T}\boldsymbol{W}_0\boldsymbol{y} \end{bmatrix}. \tag{9.30}$$

The index 0 stands using the block-diagonal matrix $\boldsymbol{\Sigma}_0$. In case model A holds we have the covariance matrix

$$\mathbb{D}(\underline{\hat{\boldsymbol{\theta}}} \mid A) = \boldsymbol{N}_0^{-1}. \tag{9.31}$$

This model is useful, since *the normal equation system can be setup point by point*, and *it will generally be sparse*, since each point only influences the coordinates of the neighbouring knots/control points of a spline surface. The sparsity of $\boldsymbol{N}_0$ has two positive numerical effects:

(i) The solution of the normal equation system can exploit the sparsity, and therefore can be performed numerically efficient.

(ii) Though the covariance matrix $\boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta},0} = \boldsymbol{N}_0^{-1}$ will be generally full, one may efficiently determine those elements of the covariance matrix, where the normal equation matrix is non-zero, without needing to determine the other elements of the covariance matrix, see Takahashi et al. (1973, cf MATLAB-code `sparseinv.m`) and Vanhatalo and Vehtari (2008).

This model certainly is too simplified, since neither possible correlations between the observations nor some, possibly available, a priori information is taken into account.

9.4.3 B: Self-calibration with fusion using independent points

In model B, we now want to fuse some a priori results $(\boldsymbol{c}_a, \boldsymbol{\Sigma}_{c_a c_a})$ within the self-calibration from (9.27). As we discussed above, this corresponds to a Bayesian estimation of the parameter vector, with prior on the calibration parameters. This then just leads to additional observations $\underline{\boldsymbol{c}}_a \sim \mathcal{M}(\boldsymbol{c}_a, \boldsymbol{\Sigma}_{c_a c_a})$ and thus the model

$$\mathbb{E}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \mid B\right) = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{I}_c \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix} \tag{9.32}$$

with the extended covariance matrix of the joint observation vector $(\boldsymbol{y}_p, \boldsymbol{c}_a)$

$$\mathbb{D}\left(\left[\begin{array}{c} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{array}\right] \mid B\right) = \left[\begin{array}{cc} \Sigma_0 & 0 \\ 0 & \Sigma_{c_a c_a} \end{array}\right]. \tag{9.33}$$

The normal equation system reads as

$$M_0 \hat{\boldsymbol{\theta}} = \boldsymbol{m}_0 \tag{9.34}$$

with

$$M_0 = N_0 + \left[\begin{array}{c} 0 \\ I_C \end{array}\right] W_{c_a c_a} [0\,,\, I_C] = \left[\begin{array}{cc} B^\mathsf{T} W_0 B & B^\mathsf{T} W_0 C \\ C^\mathsf{T} W_0 B & C^\mathsf{T} W_0 C + W_{c_a c_a} \end{array}\right] \tag{9.35}$$

and similarly

$$\boldsymbol{m}_0 = \boldsymbol{n}_0 + \left[\begin{array}{c} \mathbf{0} \\ W_{c_a c_a} \boldsymbol{c}_a \end{array}\right]. \tag{9.36}$$

Also here, the normal equation matrix will be sparse, allowing to increase numerical efficiency, both during the solution as well as for determining the covariance matrix of the parameters. This is the main motivation for using this model.

Remark: 1. Though this model formally is correct, in the context of in-situ self-calibration it contains a contradiction: The prior calibration result $(\boldsymbol{c}_s, \Sigma_{c_a c_a}$ is used explicitely, but the observed points are assumed to be uncorrelated, though they are assume to be measured by the same instrument, thus should be treated as mutually dependent. ◇

Remark: 2. In case parameters $\boldsymbol{c}$ are partitioned, namely $\boldsymbol{c}_{a2}$ of $\boldsymbol{c}_a$, e.g., if

$$\boldsymbol{c}_a = \left[\begin{array}{c} \boldsymbol{c}_{a1} \\ \boldsymbol{c}_{a2} \end{array}\right] \quad \text{and} \quad \boldsymbol{y}_c = [C_1\,,\, C_2] \left[\begin{array}{c} \boldsymbol{c}_{a1} \\ \boldsymbol{c}_{a2} \end{array}\right] = C_1 \boldsymbol{c}_{a1} + C_2 \boldsymbol{c}_{a2} \tag{9.37}$$

where the parameters $\boldsymbol{c}_{a1}$ are just fixed values, used for correcting the observations, then we can rewrite the model as

$$(\underline{\boldsymbol{y}}_p + C_1 \boldsymbol{c}_{a1} + C_2 \boldsymbol{c}_{a2}) + \underline{\boldsymbol{v}} = B\boldsymbol{y} + C\boldsymbol{c}\,, \tag{9.38}$$

Now, since the effect of the parameters $\boldsymbol{c}_{a2}$ onto the observations is the same as those of $\boldsymbol{c}$, the coefficient matrices C and C_2 coincide, why we obtain the model

$$(\underline{\boldsymbol{y}}_p + C_1 \boldsymbol{c}_{a1}) + \underline{\boldsymbol{v}} = B\boldsymbol{y} + C(\boldsymbol{c} - \boldsymbol{c}_{a2})\,, \tag{9.39}$$

Hence if we only are able to estimate the difference $\Delta\boldsymbol{c} = \boldsymbol{c} - \boldsymbol{c}_{a2}$, i.e., for given $\boldsymbol{c}_{a2}$ the corrections $\Delta\boldsymbol{c}$. ◇

Though this model takes into account the result of a prior calibration it still assumes a too simplistic covariance matrix Σ_0 for the observations, thus is statistically suboptimal, in case correlations between the points exist.

9.4.4 C: Self-calibration exploiting a priori calibration

9.4.4.1 The model

In model C, we instead of fusing the result of the a priori calibration with the current measurements, we correct take into account that the observations due to the uncertainty of the a priori calibration are correlated. Hence, we have the same functional model as in case A,

$$\mathbb{E}(\underline{\boldsymbol{y}} \mid C) = \mathbb{E}(\underline{\boldsymbol{y}}_p + C_a \underline{\boldsymbol{c}}_a \mid C) = X\boldsymbol{\theta} \quad \text{with} \quad X = [B, C]\,, \quad \boldsymbol{\theta} = \left[\begin{array}{c} \boldsymbol{y} \\ \boldsymbol{c} \end{array}\right], \tag{9.40}$$

but now assume the covariance matrix of the observations is

$$\mathbb{D}(\underline{\boldsymbol{y}} \mid C) = \Sigma_{yy,p} + C_a \Sigma_{c_a c_a} C_a^\mathsf{T} =: \Sigma\,. \tag{9.41}$$

Observe, we generally do not enforce, the self-calibration determines/corrects the same parameters as the a priori calibration, which is reasonable, in case we only want to improve the results of some calibration parameters.

Though the design matrix B is sparse, the resulting normal equation system will not be sparse. Thus – without further constraints – no numerically efficient solution is possible.

9.4.4.2 Exploiting Rao's result

This changes, if we assume the two matrices C and C_a coincide. Then Rao's lemma can be applied.

If we refer to (9.7), then, when assuming

$$C \equiv C_a\,, \quad \Sigma_0 = \Sigma_{yy,p}\,, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_{c_a c_a} \end{bmatrix} \quad \text{and} \quad \mathbf{\Theta} = 0\,, \tag{9.42}$$

we obtain $\Sigma := \Sigma_{yy}$ of (9.15), and therefore can conclude: *under the mentioned conditions, using the block-diagonal matrix $\Sigma_{yy,p}$ during estimation leads to the same estimates as when using the full covariance matrix Σ_{yy}.*

Explicitly, the estimated parameters following from the model

$$\underline{\boldsymbol{y}} + \underline{\boldsymbol{v}} = [B, C] \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix}\,, \quad \text{and} \quad \Sigma := \Sigma_{yy,p} + C\Sigma_{c_a c_a} C^{\mathsf{T}}\,. \tag{9.43}$$

are identical to those following from model A

$$\underline{\boldsymbol{y}} + \underline{\boldsymbol{v}} = [B, C] \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix}\,, \quad \text{and} \quad \Sigma_0 := \Sigma_{yy,p}\,. \tag{9.44}$$

independent on whether we correct the observations for their calibration errors $C\boldsymbol{c}_a$, as discussed above, thus

$$\boxed{\hat{\boldsymbol{\theta}} \mid C = \hat{\boldsymbol{\theta}} \mid A} \tag{9.45}$$

The covariance matrix of the estimates now results from

$$\hat{\underline{\boldsymbol{\theta}}} = (X^{\mathsf{T}} W_0 X)^{-1} X^{\mathsf{T}} W_0\, \underline{\boldsymbol{y}} \tag{9.46}$$

We obtain the uncertainty of the parameters by variance propagation as

$$\begin{aligned} \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) &= (X^{\mathsf{T}} W_0 X)^{-1} X^{\mathsf{T}} W_0 (\Sigma_0 + C\Sigma_{c_a c_a} C^{\mathsf{T}}) W_0 X (X^{\mathsf{T}} W_0 X)^{-1} && (9.47)\\ &= (X^{\mathsf{T}} W_0 X)^{-1} + (X^{\mathsf{T}} W_0 X)^{-1} X^{\mathsf{T}} W_0 C \Sigma_{c_a c_a} C^{\mathsf{T}} W_0 X (X^{\mathsf{T}} W_0 X)^{-1} && (9.48) \end{aligned}$$

But since

$$(X^{\mathsf{T}} W_0 X)^{-1} [X^{\mathsf{T}} W_0 B\,,\, X^{\mathsf{T}} W_0 C] = \left[\begin{bmatrix} I_Y \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ I_C \end{bmatrix} \right] \tag{9.49}$$

we arrive at

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) = (X^{\mathsf{T}} W_0 X)^{-1} + \begin{bmatrix} 0 \\ I_C \end{bmatrix} \Sigma_{c_a c_a} [0\,, I_C] \tag{9.50}$$

or

$$\boxed{\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) = \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) + \begin{bmatrix} 0 \\ I_C \end{bmatrix} \Sigma_{c_a c_a} [0\,, I_C]} \tag{9.51}$$

or explicitely

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) = \begin{bmatrix} \Sigma_{\hat{y}\hat{y},0} & \Sigma_{\hat{y}\hat{c},0} \\ \Sigma_{\hat{c}\hat{y},0} & \Sigma_{\hat{c}\hat{c},0} + \Sigma_{c_a c_a} \end{bmatrix} \tag{9.52}$$

Hence, using model C allows arrive at computational efficient estimation of the parameters, as well determination of their covariance matrix.

This observation gives some insight into the ability of this model to compensate for information of some prior calibration, which allows to exploit Rao's result to increase computational efficiency for determining the parameters.

However, this observation also indicates that the prior information is not fully integrated.

9.4.5 D: Self-calibration with fusion using correlated points

Model D now integrates the prior calibration and the in-situ measurements in a Bayesian self-calibration. The model now is

$$\mathbb{E}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \mid D\right) = \underbrace{\begin{bmatrix} B & C \\ 0 & I \end{bmatrix}}_{X} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix} \tag{9.53}$$

We now have to take into account that the observations $\underline{\boldsymbol{y}}$ and the prior values $\underline{\boldsymbol{c}}_a$ are correlated and use the joint covariance matrix from (9.22):

$$\begin{aligned} \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \mid D\right) &= \begin{bmatrix} \Sigma_{yy,p} + C_a \Sigma_{c_a c_a} C_a^{\mathsf{T}} & C_a \Sigma_{c_a c_a} \\ \Sigma_{c_a c_a} C_a^{\mathsf{T}} & \Sigma_{c_a c_a} \end{bmatrix} && (9.54) \\ &= \begin{bmatrix} \Sigma_{yy,p} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} C_a \\ I \end{bmatrix} \Sigma_{c_a c_a} [C_a^{\mathsf{T}}, I]\,. && (9.55) \end{aligned}$$

This model appears to enables the invocation of to Rao's Lemma if $C = C_a$, since the second column of X is common to the functional and the stochastical model.

The model then leads to the same estimated parameters as when using the covariance matrix

$$\mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{yy,p} & 0 \\ 0 & 0 \end{bmatrix}, \tag{9.56}$$

However, this implies, that the parameters $\boldsymbol{c}_a$ from the prior calibration have variance zero, thus are taken as fixed values, just correcting the observations $\boldsymbol{y} - C\boldsymbol{c}_a$ in *a non-self-calibrating model*

$$\boldsymbol{y} - C\boldsymbol{c}_a = B\boldsymbol{y} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma_0\,. \tag{9.57}$$

Again, using the actual (fully populated) covariance matrix of the joint observation vector into consideration, we obtain the covariance matrix of the estimates by variance propagation from

$$\begin{aligned} \begin{bmatrix} \underline{\hat{\boldsymbol{y}}} \\ \underline{\hat{\boldsymbol{c}}} \end{bmatrix} &= \begin{bmatrix} (B^{\mathsf{T}} W_0 B)^{-1} B^{\mathsf{T}} W_0 & 0 \\ 0 & I_C \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{y}} - C\underline{\boldsymbol{c}}_a \\ \underline{\boldsymbol{c}}_a \end{bmatrix} && (9.58) \\ &= \begin{bmatrix} (B^{\mathsf{T}} W_0 B)^{-1} B^{\mathsf{T}} W_0 & -(B^{\mathsf{T}} W_0 B)^{-1} B^{\mathsf{T}} W_0 C \\ 0 & I_C \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix}. && (9.59) \end{aligned}$$

Using

$$N_{11} = B^{\mathsf{T}} W_0 B \quad \text{and} \quad N_{12} = B^{\mathsf{T}} W_0 C \tag{9.60}$$

neglecting the index zero, this reads as

$$\begin{bmatrix} \underline{\hat{\boldsymbol{y}}} \\ \underline{\hat{\boldsymbol{c}}} \end{bmatrix} = \begin{bmatrix} N_{11}^{-1} B^{\mathsf{T}} W_0 & -N_{11}^{-1} N_{12} \\ 0 & I_C \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{c}}_a \end{bmatrix} \tag{9.61}$$

and we obtain

$$\begin{aligned} \mathbb{D}(\underline{\hat{\boldsymbol{\theta}}} \mid D) &= \begin{bmatrix} N_{11}^{-1} B^{\mathsf{T}} W_0 & -N_{11}^{-1} N_{12} \\ 0 & I_C \end{bmatrix} \begin{bmatrix} \Sigma_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W_0 B N_{11}^{-1} & 0 \\ -N_{21} N_{11}^{-1} & I_C \end{bmatrix} && (9.62) \\ &\quad + \begin{bmatrix} N_{11}^{-1} B^{\mathsf{T}} W_0 & -N_{11}^{-1} N_{12} \\ 0 & I_C \end{bmatrix} \begin{bmatrix} C \\ I \end{bmatrix} \Sigma_{c_a c_a} [C^{\mathsf{T}}, I] \begin{bmatrix} W_0 B N_{11}^{-1} & 0 \\ -N_{21} N_{11}^{-1} & I_C \end{bmatrix} \\ &= \begin{bmatrix} (B^{\mathsf{T}} W_0 B)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_C \end{bmatrix} \Sigma_{c_a c_a} [0, I_C] && (9.63) \\ &= \begin{bmatrix} (B^{\mathsf{T}} W_0 B)^{-1} & 0 \\ 0 & \Sigma_{c_a c_a} \end{bmatrix}. && (9.64) \end{aligned}$$

This is a stunning result: The fusion of the prior information $(\boldsymbol{c}_a, \Sigma_{c_a c_a})$ from the calibration does neither improve the calibration parameters, the correlations assumed for the joint observation vector $(\boldsymbol{y}, \boldsymbol{c}_e)$ (1) have no effect onto the calibration parameters, (2) are not needed to determine the object parameters $\boldsymbol{y}$, and (3) decorrelate the estimation of the parameter for object and calibration.

9.5 Synopsis

The following Table 9.2 collects the results, especially the covariance matrices

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid k)\,, \quad k = A, B, C, D\,. \tag{9.65}$$

for the four cases.

	$\mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma_0$		$\mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma$
SC	$\hat{\boldsymbol{\theta}} \mid A$	$\equiv$	$\hat{\boldsymbol{\theta}} \mid C$
	$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) = \begin{bmatrix} B^{\mathsf{T}}W_0B & B^{\mathsf{T}}W_0C \\ C^{\mathsf{T}}W_0B & C^{\mathsf{T}}W_0C \end{bmatrix}^{-1}$ (9.29)		$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) = \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) + \begin{bmatrix} 0 \\ I_C \end{bmatrix} \Sigma_{c_a c_a} [0\,, I_C]$ (9.35)
BSC	$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid B) = \begin{bmatrix} B^{\mathsf{T}}W_0B & B^{\mathsf{T}}W_0C \\ C^{\mathsf{T}}W_0B & C^{\mathsf{T}}W_0C + W_{c_a c_a} \end{bmatrix}^{-1}$ (9.51)		$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid D) =$ D: $\begin{bmatrix} (B^{\mathsf{T}}W_0B)^{-1} & 0 \\ 0 & \Sigma_{c_a c_a} \end{bmatrix}$ (9.64)

Table 9.2: The covariance matrices of the estimated parameters when using the four models for self-calibration, SC: self-calibration without prior, BSC: Bayesian self-calibration

First, the estimated parameters for model A and C are the same, see (9.45):

$$\hat{\boldsymbol{\theta}} \mid C = \hat{\boldsymbol{\theta}} \mid A\,. \tag{9.66}$$

Second, we compare the accuracy achievable in the different models:

1. the influence of changing the covariance matrix onto the accuracy can be determined for models A and C. Since the uncertainty of the observations in model A are assumed to be not larger than that in model C, hence because $\Sigma - \Sigma_0 \geq 0$ the uncertainty of the parameters in model C generally is larger than that of models A:

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid C) \geq \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A)\,. \tag{9.67}$$

   However, the accuracy of the object parameters for models A and C is the same:

$$\mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid C) = \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid A)\,. \tag{9.68}$$

2. the influence of the fusion of prior and in-situ self-calibration can be determined for models A and B. Since the model B includes additional, independent information compared to model A, the uncertainty generally increased by the fusion process:

$$\mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid A) \geq \mathbb{D}(\hat{\underline{\boldsymbol{\theta}}} \mid B)\,. \tag{9.69}$$

   Observe, this holds for both, the parameters $\boldsymbol{y}$ of the object as well as the calibration parameters $\boldsymbol{c}$, which easily can be seen using the Schur complements of the two diagonal block matrices of the covariance matrices.

3. the accuracy of the estimated parameters in model D cannot be compared to the others in general, since it is not a generalization of one of them. However, the accuracy of the estimated object parameters can be compared. We especially have

$$\mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid D) \leq \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid B) \leq \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid A) = \mathbb{D}(\hat{\underline{\boldsymbol{y}}} \mid C)\,, \tag{9.70}$$

   again using the Schur complements of the corresponding covariance matrices.

9.6 Concluding remarks

Generally, these result only are valid, if Rao's lemma can be applied, i.e., if the calibration parameters $\boldsymbol{c}$ determined in the self-calibration are the same which cause the correlations between the points, formally if the coefficient matrix C in the functional model is the same as the one C_a used in the stochastical model, hence if $C_a = C$. This may, be enforced by assuming the calibration parameters not corrected in the self-calibration have zero effect onto the observed points, e.g., of one assumes the these parameters, which are determined in the prior calibration, have small enough variance, to assume it to be zero.

9.7 Appendix: Covariance matrix for given design matrix, observations, estimate and covariance matrix of parameters

On can show, that there is a set of covariance matrices Σ_{yy} if the following is given:

1. the linear model $\mathbb{E}(\underline{y}) = A\boldsymbol{\theta}$,
2. the value of the estimate and its covariance matrix $\{(\widehat{\boldsymbol{\theta}}, \Sigma_{\widehat{\theta}\widehat{\theta}}) = (\boldsymbol{\theta}, V)\}$ of the parameters, and
3. a vector $\boldsymbol{y}$ of observations,

such that the estimated parameters and their covariance matrix follow from a weighted least squares estimation.

9.7.1 Example: The mean of two values $y_i, i = 1, 2$

Given are two observations $\boldsymbol{y} = [y_i]$ and an estimate $\theta = \widehat{\theta}$ for the mean with variance $v = \sigma^2_{\widehat{\theta}}$. The covariance matrix of the observations is to be chosen adequately.

9.7.1.1 A special solution

We have the following model

$$\mathbb{E}(\underline{\boldsymbol{y}}) = A\theta \quad \text{with} \quad A = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{1}_2 \tag{9.71}$$

and need to choose, say in the form, containing the correlation coefficient $\rho \in [-1, +1]$

$$\mathbb{D}(\underline{\boldsymbol{y}}) = \sigma^2 \begin{bmatrix} 1 & \rho k \\ \rho k & k^2 \end{bmatrix} \quad \text{with} \quad \sigma_{y_1} = \sigma \quad \text{and} \quad \sigma_{y_2} = k\sigma\,, \tag{9.72}$$

such that the two constraints

$$\theta = \widehat{\theta} = (A^{\mathsf{T}}\Sigma_{yy}^{-1}A)^{-1}\, A^{\mathsf{T}}\Sigma_{yy}^{-1}\boldsymbol{y}\,, \tag{9.73}$$

$$v = \sigma^2_{\widehat{\theta}} = (A^{\mathsf{T}}\Sigma_{yy}^{-1}A)^{-1}\,. \tag{9.74}$$

This are two constraints for the three not yet specified parameters σ, k, and ρ.

Explicitely, we obtain

$$W_{yy} = \frac{1}{k^2\sigma^2(1-\rho^2)} \begin{bmatrix} k^2 & -\rho k \\ -\rho k & 1 \end{bmatrix} \tag{9.75}$$

and

$$\widehat{\theta} = \frac{l_2 + k^2\, l_1 - k\, l_1\, \rho - k\, l_2 \rho}{k^2 - 2\rho k + 1} = \frac{k(k-\rho)l_1 - (k\rho - 1)l_2}{k^2 - 2\rho k + 1} \tag{9.76}$$

$$\widehat{\sigma}^2_{\widehat{\theta}} = \frac{k^2(1-\rho^2)}{k^2 - 2\rho k + 1}\,\sigma^2 \tag{9.77}$$

From the two constraints

$$\hat{\theta} = \theta \quad \text{and} \quad \hat{\sigma}^2_{\hat{\theta}} = v \tag{9.78}$$

we obtain the two parameters σ^2 and ρ as a function of k and the given observations:

$$\begin{aligned}
\sigma^2 &= \frac{v\ (l_1 - l_2)\ (l_1 + l_2 - 2\,x)}{k^2\,{l_1}^2 - 2\,k^2\,l_1\,x + k^2\,x^2 - {l_2}^2 + 2\,l_2\,x - x^2} && (9.79)\\
&= \frac{v\ (l_1 - l_2)\ (l_1 + l_2 - 2\,x)}{((l_2 - x) + kl_1 - kx)(-(l_2 - x) + kl_1 - kx)} && (9.80)\\
\rho &= \frac{l_2 - x + k^2\,l_1 - k^2\,x}{k\ (l_1 + l_2 - 2\,x)} && (9.81)\\
&= \frac{(l_2 - k^2 l_1) + (1 - k^2)x}{k\ (l_1 + l_2 - 2\,x)} && (9.82)
\end{aligned}$$

For the special case

$$l_1 = 1\,, \quad l_2 = 0\,, \quad x = -1 \tag{9.83}$$

we obtain

$$\sigma^2 = \frac{3}{4k^2 - 1}v \quad \text{and} \quad \rho = \frac{2k^2 + 1}{3k} \quad \text{for} \quad k \in (0.5, 1) \tag{9.84}$$

9.7.1.2 A generalizable solution

We use the following Fig. 9.3, assuming $\boldsymbol{x} = X = \mathbf{1}_2$, and we observe the following:

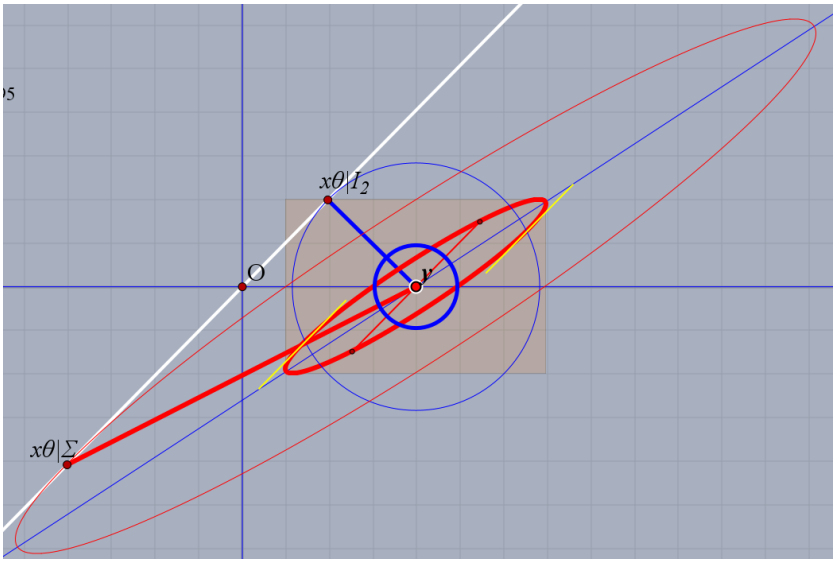


Figure 9.3: The generalized mean

1. The observed point $\boldsymbol{y}$ is slantly projected to $\boldsymbol{x}\hat{\theta}|\boldsymbol{\Sigma}$ on the line $\boldsymbol{x}\theta$.
2. The length of radius of the standard ellipse parallel to the line leads to the standard interval of the estimated point $\boldsymbol{x}\hat{\theta}$, in the figure half of length of the yellow tangent segment.
3. The length of the conjugate diameter is irrelevant for both, the position and the standard deviation of the estimate.

Hence we can specify the set of covariance matrices by mapping the reference covariance matrix $\boldsymbol{\Sigma}_0 = \boldsymbol{I}_2$ to $\boldsymbol{\Sigma}$ by applying the mapping the two unit vectors $\boldsymbol{e}_i, i = 1, 2$ to the tow

conjugate diameters $\boldsymbol{d}_i, i = 1, 2$ of the standard ellipse of $\boldsymbol{\Sigma}$. The two conjugate diameters are

$$\boldsymbol{d}_1 = \frac{\boldsymbol{x}}{|\boldsymbol{x}|}\sigma \quad \text{and} \quad \boldsymbol{d}_2(f) = \frac{\boldsymbol{x}\theta - \boldsymbol{y}}{|\boldsymbol{x}\theta - \boldsymbol{y}|}f \text{ for some arbitrary standard deviation} f > 0\,. \tag{9.85}$$

Hence we obtain the set of covariance matrices, parametrized by f from

$$\begin{aligned}
\boldsymbol{\Sigma}(f) &= [\boldsymbol{d}_1\ \boldsymbol{d}_2(f)] \begin{bmatrix} \boldsymbol{d}_1^\mathsf{T} \\ \boldsymbol{d}_2^\mathsf{T}(f) \end{bmatrix} = \boldsymbol{d}_1\boldsymbol{d}_1^\mathsf{T} + \boldsymbol{d}_2(f)\boldsymbol{d}_2^\mathsf{T}(f) && (9.86)\\
&= \frac{\boldsymbol{x}\boldsymbol{x}^\mathsf{T}}{\boldsymbol{x}^\mathsf{T}\boldsymbol{x}}\sigma^2 + \frac{(\boldsymbol{x}\theta - \boldsymbol{y})(\boldsymbol{x}\theta - \boldsymbol{y})^\mathsf{T}}{(\boldsymbol{x}\theta - \boldsymbol{y})^\mathsf{T}(\boldsymbol{x}\theta - \boldsymbol{y})}f^2 && (9.87)\\
&= \frac{\boldsymbol{x}\boldsymbol{x}^\mathsf{T}}{\boldsymbol{x}^\mathsf{T}\boldsymbol{x}}\sigma^2 + \frac{\boldsymbol{r}\boldsymbol{r}^\mathsf{T}}{\boldsymbol{r}^\mathsf{T}\boldsymbol{r}}f^2 \quad \text{with} \quad \boldsymbol{r} = \boldsymbol{x}\theta - \boldsymbol{y} && (9.88)
\end{aligned}$$

10 The Mean of Correlated Observations

For uncorrelated observations the accuracy of the mean increases with the number of observations. In case they are correlated, there is an upper limit for the accuracy. The note analyses the situation for constant correlation and for exponentially decaying correlation, autoregressive noise.

10.1 Preface

The arithmetic mean in many cases can be used as a proxy for a more general estimation problem. Here, we analyse the effect of correlations onto the accuracy of the estimated mean. The Note 11 generalizes the results.

10.2 Goal

We derive the precision of the correlated mean, by generatively model the observed values as a mean value which additively is distorted by a random effect with zero mean. We discuss two cases:

1. The noise in the measurements $\underline{y}_i$ consists of a uncorrelated part $\underline{d}_i$ and a correlated part $\underline{b}$, where the correlated part $\underline{b}$ is describable as a noisy bias. Namely, we have:

$$\underline{y}_i = \mu + \underline{b} + \underline{d}_i\,, \quad i = 1, ..., N \quad \text{with} \quad \underline{d}_i \sim \mathcal{M}(0, \sigma_d^2) \quad \text{and} \quad \underline{b} \sim \mathcal{M}(0, \sigma_b^2) \tag{10.1}$$

   We obtain the following result:

   (a) The estimated mean is independent of the bias

$$\hat{\mu} = \frac{1}{N} \sum_n y_n\,. \tag{10.2}$$

   (b) The variance of the estimated mean cannot become smaller than the variance of the bias

$$\sigma_{\hat{\mu}}^2 = \frac{\sigma_d^2}{N} + \sigma_b^2\,. \tag{10.3}$$

2. The noise is an autoregressive process of first order AR(1), namely

$$\underline{y}_i = \mu + \underline{b}_i\,, \quad n = 1, ..., N \tag{10.4}$$

and the AR(1)-process with parameter a

$$b_i = ab_{i-1} + \underline{e}_i \quad \text{with} \quad n > 1 \tag{10.5}$$

starting with

$$\underline{e}_i \sim \mathcal{M}(0, \sigma_e^2) \quad \text{and} \quad \underline{e}_1 = \mathcal{M}\left(0, \frac{\sigma_e^2}{1-a^2}\right) \tag{10.6}$$

The variance of the estimated mean is

$$\sigma_{\hat{\mu}}^2 = \frac{1+a}{1+(1-2/N)a}\frac{\sigma_e^2}{N} \quad \text{with} \quad |a| < 1\,. \tag{10.7}$$

10.3 Random constant bias

10.3.1 The model

We can write the generative model as

$$\underline{\boldsymbol{y}} = \mathbf{1}\mu + \mathbf{1}\underline{b} + \underline{\boldsymbol{n}} \tag{10.8}$$

leading to the covariance matrix

$$\boldsymbol{\Sigma}_{yy} = \sigma_d^2 \boldsymbol{I}_N + \mathbf{1}\mathbf{1}^\mathsf{T}\sigma_b^2\,. \tag{10.9}$$

Thus, the observed values have the variance and covariance

$$\sigma_{y_i}^2 = \sigma_d^2 + \sigma_b^2 \quad \text{and} \quad \sigma_{y_i y_j} = \sigma_b^2 \text{ for } i \neq j \tag{10.10}$$

hence have correlation coefficient

$$\rho_{ij} = \frac{\sigma_b^2}{\delta_{ij}\sigma_d^2 + \sigma_b^2} > 0 \text{ for } i \neq j\,. \tag{10.11}$$

Therefore, with the common correlation coefficient

$$\rho = \frac{\sigma_b^2}{\sigma_d^2 + \sigma_b^2} \text{ for } i \neq j\,, \tag{10.12}$$

the covariance matrix explicitly reads

$$\boldsymbol{\Sigma}_{yy} = (\sigma_d^2 + \sigma_b^2)\begin{bmatrix} 1 & \dots & \rho & \dots & \rho \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho & \dots & 1 & \dots & \rho \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho & \dots & \rho & \dots & 1 \end{bmatrix}. \tag{10.13}$$

10.3.2 The solution

The Gauss-Markov model reads

$$\boldsymbol{y} + \boldsymbol{v} = \mathbf{1}\mu \quad \text{with} \quad \boldsymbol{\Sigma}_{ll} = \sigma_d^2 \boldsymbol{I}_N + \sigma_b^2 \mathbf{1}\mathbf{1}^\mathsf{T} \tag{10.14}$$

The weight matrix of the observations has the structure

$$\boldsymbol{W}_{ll} = a\boldsymbol{I}_N + b\mathbf{1}\mathbf{1}^\mathsf{T}\,. \tag{10.15}$$

Therefore, we can determine a and b from

$$(\sigma_d^2 \boldsymbol{I} + \sigma_b^2 \mathbf{1}\mathbf{1}^\mathsf{T})(a\boldsymbol{I} + b\mathbf{1}\mathbf{1}^\mathsf{T}) = \underbrace{a\sigma_d^2}_{=1} \boldsymbol{I} + \underbrace{(a\sigma_b^2 + b\sigma_d^2 + bN\sigma_b^2)}_{!=0} \mathbf{1}\mathbf{1}^\mathsf{T} . \tag{10.16}$$

We obtain

$$a = \frac{1}{\sigma_d^2} \quad \text{and} \quad b = -\frac{a\sigma_b^2}{\sigma_d^2 + N\sigma_b^2} = -\frac{\sigma_b^2}{\sigma_d^2(\sigma_d^2 + N\sigma_b^2)} \tag{10.17}$$

Therefore the weight matrix is

$$\boldsymbol{W}_{yy} = \frac{1}{\sigma_d^2}\boldsymbol{I}_N - \frac{\sigma_b^2}{\sigma_d^2(\sigma_d^2 + N\sigma_b^2)}\mathbf{1}\mathbf{1}^\mathsf{T} = \frac{1}{\sigma_d^2}\left(\boldsymbol{I} - \frac{\sigma_b^2}{\sigma_d^2 + N\sigma_b^2}\mathbf{1}\mathbf{1}^\mathsf{T}\right) \tag{10.18}$$

The normal equation system is

$$\boldsymbol{N}\hat{\boldsymbol{\theta}} = \boldsymbol{n} \tag{10.19}$$

with

$$\boldsymbol{N} = \mathbf{1}^\mathsf{T}\boldsymbol{W}_{ll}\mathbf{1} = \frac{N}{\sigma_d^2} - \frac{N^2\sigma_b^2}{\sigma_d^2(\sigma_d^2 + N\sigma^2)} = \frac{N}{\sigma_d^2}\left(1 - \frac{N\sigma_b^2}{\sigma_d^2 + N\sigma_b^2}\right) \tag{10.20}$$

and

$$\boldsymbol{n} = \mathbf{1}^\mathsf{T}\boldsymbol{W}_{ll}\,\boldsymbol{y} = \frac{1}{\sigma_d^2}\left(\mathbf{1}^\mathsf{T} - \frac{N\sigma_b^2}{\sigma_d^2 + N\sigma_b^2}\mathbf{1}^\mathsf{T}\right)\boldsymbol{y} = \frac{1}{\sigma_d^2}\left(1 - \frac{N\sigma_b^2}{\sigma_d^2 + N\sigma^2}\right)\sum_n y_n . \tag{10.21}$$

The solution for the mean is

$$\hat{\mu} = \frac{1}{N}\sum_n y_n . \tag{10.22}$$

Hence we have the result: **The correlated arithmetic mean is independent on the correlation coefficient.**

The variance of the estimated mean is

$$\sigma_{\hat{\mu}}^2 = \frac{\sigma_d^2}{N}\frac{\sigma_d^2 + N\sigma_b^2}{\sigma_d^2} = \frac{\sigma_d^2}{N} + \sigma_b^2 \tag{10.23}$$

Hence we have the result: **The variance of the estimated mean of correlated observations diminishes with increasing N but cannot be smaller than the variance of the bias.** Hence, in case the variance of the bias is much larger than the variance of the noise, the variance of the mean is close to the variance of the bias.

10.3.3 Alternative derivation

We assume the model

$$\begin{bmatrix} \boldsymbol{y} \\ b_0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{v} \\ v_{b_0} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ b \end{bmatrix} \quad \text{with} \quad \mathbb{D}\left(\begin{bmatrix} \boldsymbol{y} \\ \underline{b}_0 \end{bmatrix}\right) = \begin{bmatrix} \sigma_d^2\boldsymbol{I} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & \sigma_b^2 \end{bmatrix} \tag{10.24}$$

with $b_0 = 0$, since we assumed $\underline{b} \sim \mathcal{M}(0, \sigma_b^2)$. The normal equation matrix

$$\boldsymbol{N} = \begin{bmatrix} \mathbf{1}^\mathsf{T} & 0 \\ \mathbf{1}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} w_d\boldsymbol{I} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & w_b \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} w_d N & w_d N \\ w_d N & w_d N + w_b \end{bmatrix} \tag{10.25}$$

with its inverse

$$\boldsymbol{N}^{-1} = \frac{1}{N w_d w_b}\begin{bmatrix} w_d N + w_b & -w_d N \\ -w_d N & w_d N \end{bmatrix} \tag{10.26}$$

Hence the variances of the estimate $\hat{\mu}$ is

$$\sigma_{\hat{\mu}}^2 = \frac{\sigma_d^2}{N} + \sigma^2 , \tag{10.27}$$

as above.

We also can directly determine the variance of $\widehat{\mu}$ using the Schur-complement of N in (10.25):

$$\sigma^2_{\widehat{\mu}} = \left(w_d N - \frac{w_d^2 N^2}{w_d N + w_b} \right)^{-1} \tag{10.28}$$

$$= \left(\frac{w_d^2 N^2 + w_d w_b N - w_d^2 N^2}{w_d N + w_b} \right)^{-1} = \frac{\sigma_d^2}{N} + \sigma_b^2 \,. \tag{10.29}$$

10.3.4 Covariance of arithmetic mean with correlated observations

The simple arithmetic mean assumes $\Sigma_0 := \sigma_d^2 I$. Then the estimate is

$$\widehat{\mu} = \frac{\mathbf{1}^\mathsf{T}}{N} \boldsymbol{y} \,, \tag{10.30}$$

with covariance matrix, assuming Σ_0 holds

$$\sigma^2_{\widehat{\mu}_0} = \frac{\sigma_d^2}{N} \,. \tag{10.31}$$

If this arithmetic mean is taken, but the actual covariance matrix is $\sigma_d^2 I_N + \sigma_b^2 \mathbf{1}\mathbf{1}^\mathsf{T}$ variance propagation of (10.30) yields

$$\sigma^2_{\widehat{\mu}} = \frac{\mathbf{1}^\mathsf{T}}{N} (\sigma_d^2 I_N + \sigma_b^2 \mathbf{1}\mathbf{1}^\mathsf{T}) \frac{\mathbf{1}}{N} \tag{10.32}$$

This can be simplified to

$$\sigma^2_{\widehat{\mu}} = \frac{\sigma_d^2}{N} + \sigma_b^2 \,, \tag{10.33}$$

which coinsides with (10.23).

10.3.5 Using a more general covariance matrix

The observations up to now have been assumed to be positively correlated, see (10.12)

$$\rho = \frac{\sigma_b^2}{\sigma_d^2 + \sigma_b^2} \quad \text{or} \quad \frac{\sigma_b^2}{\sigma_d^2} = \frac{\rho}{1-\rho} \,. \tag{10.34}$$

However, they also may have negative correlation. Of course, this then cannot be explained by a stochastic bias term anymore.

Therefore we assume

$$\Sigma_{yy} = \sigma_l^2 \begin{bmatrix} 1 & \dots & \rho & \dots & \rho \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho & \dots & 1 & \dots & \rho \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho & \dots & \rho & \dots & 1 \end{bmatrix} = \sigma_l^2 (1-\rho) I_N + \sigma_l^2 \rho \mathbf{1}\mathbf{1}^\mathsf{T} \,. \tag{10.35}$$

If $\rho < 0$ this is implicitly assuming $\sigma_b^2 < 0$. Therefore, we use the derivation above, which is valid also for $\sigma_b^2 < 0$. We now realize, that the correlation cannot have an arbitrary negative value ≥ -1, since the variance (10.23) of the mean needs to be positive. This leads to the following constraint, first formally

$$\frac{\sigma_d^2}{N} \geq \sigma_b^2 \quad \text{or} \quad \frac{1}{N} \geq \frac{\sigma_b^2}{\sigma_d^2} \tag{10.36}$$

then using (10.34)

$$\frac{1}{N} \geq \frac{\rho}{1-\rho} \quad \text{or} \quad \frac{\rho}{\rho-1} \geq \frac{-1}{N}\,, \tag{10.37}$$

which finally leads to a constraint on the correlation coefficient

$$\rho \geq \frac{-1}{N-1}\,. \tag{10.38}$$

For example: for two observations $N = 2$ the correlation coefficient may be arbitrary in the range $[-1, +1]$, but for three observations $N = 3$ the correlation coefficient needs to be larger than -50%.

10.4 Random autoregressive noise

10.4.1 The model

The N observations result from

$$\underline{\boldsymbol{y}} = \mathbf{1}\mu + \underline{\boldsymbol{b}} \tag{10.39}$$

with the following covariance matrices

$$\mathbb{D}(\underline{\boldsymbol{b}}) = \Sigma_{bb} = [\Sigma_{b_i b_j}] = \frac{\sigma_e^2}{1-a^2}\left[a^{|i-j|}\right] \quad \text{with} \quad |a| < 1\,. \tag{10.40}$$

Hence the covariance matrix of the noise is

$$\Sigma_{yy} = \Sigma_{bb} \tag{10.41}$$

We have the inverse of the covariance matrix of the bias, which is a tridiagonal matrix:

$$W_{yy} = \Sigma_{bb}^{-1} = \frac{1}{\sigma_e^2}\,\mathrm{Tri}[1, 1+a^2, ..., 1+a^2, 1][-a, ..., -a] \tag{10.42}$$

10.4.2 The solution

The Gauss–Markov model reads as

$$\hat{\boldsymbol{y}} + \boldsymbol{v} = \mathbf{1}\mu \quad \text{with} \quad \Sigma_{yy} = \Sigma_{bb} = W_{bb}^{-1}\,. \tag{10.43}$$

The normal equation system now is

$$N\hat{\boldsymbol{\theta}} = \boldsymbol{n} \tag{10.44}$$

with (canceling the common factor $\sigma_e^2/(1-a)$)

$$N = \mathbf{1}^\mathsf{T} W_{yy}\mathbf{1}\,\frac{\sigma_e^2}{1-a} = \frac{2 + (N-2)(1+a^2) - 2(N-1)a}{1-a} = (N - (N-2)a) \tag{10.45}$$

and

$$\boldsymbol{n} = \mathbf{1}^\mathsf{T} W_{yy}\boldsymbol{x}\,\frac{\sigma_e^2}{1-a} = \sum_{n=1}^{N} x_i - a\sum_{i=2}^{N-1} x_i\,. \tag{10.46}$$

The variance of the estimated mean is

$$\sigma_{\hat{\mu}}^2 = \frac{1}{(1-a)(N-(N-2)a)}\sigma_e^2 \tag{10.47}$$

which can be rewritten as

$$\sigma_{\hat{\mu}}^2 = \frac{1}{(1-a)(1-(1-2/N)a)}\,\frac{\sigma_e^2}{N}\,. \tag{10.48}$$

We have the following limits.

- For large N we achieve

$$\lim_{N\to\infty} \sigma^2_{\hat{\mu}} = \frac{1}{1-a^2}\frac{\sigma_e^2}{N}\,, \tag{10.49}$$

  thus the variance is larger by a factor $1/(1-a^2)$ compared to the uncorrelated mean.
- For $a = 0$ we obtain the result of the uncorrelated mean.
- For $a = 1$, the noise process is semi-stationary, and we obtain

$$\lim_{a\to 1} \sigma^2_{\hat{\mu}} = \infty \tag{10.50}$$

  independent on the number of observations.

10.5 The general case

The situation of the mean with constant correlation is a special case discussed in Rao (1967) in Lemma 5a. The estimated parameters of the model $(\boldsymbol{y} = X\boldsymbol{\theta}, \Sigma)$ are the same if instead of Σ the covariance matrix $\Sigma + X\boldsymbol{\Gamma}X^\mathsf{T}$ is used. In our case we used $\boldsymbol{\Gamma} = \sigma^2$. The efficiency of the estimate, thought being the same, is reduced due to the correlations induced by $\boldsymbol{b}$.

Generalizing (10.24), we use the model, assuming $\boldsymbol{\Gamma} = \Sigma_{bb}$,

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{b}_0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{v}_b \end{bmatrix} = \begin{bmatrix} X & X \\ 0^\mathsf{T} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{b} \end{bmatrix} \quad \text{with} \quad \mathbb{D}\left(\begin{bmatrix} \boldsymbol{y} \\ \underline{\boldsymbol{b}}_0 \end{bmatrix} \right) = \begin{bmatrix} \Sigma_{yy} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & \Sigma_{bb} \end{bmatrix} \tag{10.51}$$

again assuming $\boldsymbol{b}_0 = \mathbf{0}$. The normal equation matrix is

$$\begin{aligned} N &= \begin{bmatrix} M & M \\ M & M + W_{bb} \end{bmatrix} && (10.52)\\ &= \begin{bmatrix} X^\mathsf{T} & 0 \\ X^\mathsf{T} & I \end{bmatrix} \begin{bmatrix} W_{yy} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & W_{bb} \end{bmatrix} \begin{bmatrix} X & X \\ \mathbf{0} & I \end{bmatrix} && (10.53)\\ &= \begin{bmatrix} X^\mathsf{T}W_{yy}X & X^\mathsf{T}W_{yy}X \\ X^\mathsf{T}W_{yy}X & X^\mathsf{T}W_{yy}X + W_{bb} \end{bmatrix} && (10.54) \end{aligned}$$

with its inverse

$$\Sigma_{\hat{p}\hat{p}} = N^{-1} = \begin{bmatrix} \Sigma_{\hat{\mu}\hat{\mu}} & \Sigma_{\hat{\mu}\hat{b}} \\ \Sigma_{\hat{b}\hat{\mu}} & \Sigma_{\hat{b}\hat{b}} \end{bmatrix}. \tag{10.55}$$

Now we use the inverse of Schur-complement,

$$\Sigma_{\hat{\mu}\hat{\mu}} = (M - M(M + W_{bb})^{-1}M)^{-1}\,, \tag{10.56}$$

and the Woodbury identity,

$$(A + CBC^\mathsf{T})^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^\mathsf{T}A^{-1}C)^{-1}C^\mathsf{T}A^{-1}\,, \tag{10.57}$$

with $A = M^{-1}$, $B = \Sigma_{bb}$ and $C = I$, and obtain

$$\Sigma_{\hat{\mu}\hat{\mu}} = M^{-1} + \Sigma_{bb} \tag{10.58}$$

hence,

$$\Sigma_{\hat{\mu}\hat{\mu}} = (X^\mathsf{T}\Sigma_{yy}^{-1}X)^{-1} + \Sigma_{bb}\,. \tag{10.59}$$

The prior, influencing all parameters the same way, leads to an increase of the covariance matrix.

Again, the result, using $\boldsymbol{\Gamma} = \Sigma_{bb}$, also holds if $\boldsymbol{\Gamma} < \mathbf{0}$, but only if

$$(X^\mathsf{T}\Sigma_{yy}^{-1}X)^{-1} \geq \Sigma_{bb} \tag{10.60}$$

11 Accuracy of the Mean when using a Wrong Covariance Matrix

Suboptimal, i.e., approximate solutions often are used or needed when estimating parameters. One of such simplifications refers to the stochastical model, especially the covariance matrix of the observations, which often is assumed to be a multiple of a unit matrix, implicitly assuming all observations have the same weight and are mutually uncorrelated. This note provides the general relation between the accuracy of the estimated parameters when using an approximate covariance matrix and exemplifies this using the mean of repeated observations.

11.1 Preface

The arithmetic mean in many cases can be used as a proxy for a more general estimation problem. Here, we analyse the loss in accuracy of the estimated mean when using a wrong covariance matrix. The note generalizes the results from Note **??**.

11.2 Summary

The note shows the effect of using a wrong covariance matrix when estimating parameters. Especially we obtain the following results for estimating the mean from N values l_n:

1. If the mean is estimated assuming, that all values have the same weight $w = 1/\sigma^2$, thus $\widehat{\theta} = \sum_n l_n/N$, but the values really have individual weights

$$w_n = \frac{1}{\tilde{\sigma}_n^2} \tag{11.1}$$

then the variance of the approximately determined mean is larger by a factor

$$\lambda = \overline{\tilde{\sigma}_n^2} \cdot \overline{1/\tilde{\sigma}_n^2} = \frac{\mu_{\tilde{\sigma}^2}^{(a)}}{\mu_{\tilde{\sigma}^2}^{(h)}} \tag{11.2}$$

thus the ratio of the arithmetic mean and the harmonic mean of the variances. The factor λ only is 1, in case the variances are identical for all values l_n.

2. In the special case, that the weights are assumed to be randomly taken from a Gamma distribution and their relative variation is

$$c = \frac{\sigma_w}{\mu_w} < 1\,, \tag{11.3}$$

then the factor is given by

$$\lambda = \frac{1}{1-c^2}\,. \tag{11.4}$$

If $c \geq 1$ the factor is not limited.

11.3 Problem

If the estimation is performed in a Gauss-Markov model $\mathbb{E}(\underline{\boldsymbol{y}}) = \boldsymbol{X\theta}$ with $\boldsymbol{\Sigma}_{yy} = \boldsymbol{\Sigma}$ but the true covariance matrix of the observations is $\widetilde{\boldsymbol{\Sigma}_{yy}} = \tilde{\boldsymbol{\Sigma}}$, then the covariance matrix of the estimated parameters is

$$\boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}} = (\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\;\tilde{\boldsymbol{\Sigma}}\;\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\,, \tag{11.5}$$

which follows from $\hat{\underline{\boldsymbol{\theta}}} = (\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\underline{\boldsymbol{y}} - \boldsymbol{x})$. Observe, only if $\boldsymbol{\Sigma} = \tilde{\boldsymbol{\Sigma}}$ do we obtain the classical result

$$\tilde{\boldsymbol{\Sigma}}_{\hat{\theta}\hat{\theta}} = (\boldsymbol{X}^\mathsf{T}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X})^{-1}\,. \tag{11.6}$$

11.4 The accuracy of the approximate solution

The relation between both covariance matrices can be derived from the eigenvalues of the quotient

$$\lambda(\boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}}\tilde{\boldsymbol{\Sigma}}^{-1}_{\hat{\theta}\hat{\theta}}) = \lambda\left((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\;\tilde{\boldsymbol{\Sigma}}\;\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\;\boldsymbol{X}^\mathsf{T}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)\,. \tag{11.7}$$

Equations (11.5) and (11.7) can be used to investigate the effect of choosing a simplified stochastical model, e.g., when using $\boldsymbol{\Sigma}_{yy} = \sigma^2\boldsymbol{I}_N$ instead of $\tilde{\boldsymbol{\Sigma}}$.

For $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$ we would obtain

$$\lambda(\boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}}\tilde{\boldsymbol{\Sigma}}^{-1}_{\hat{\theta}\hat{\theta}}) = \lambda\left((\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\;\tilde{\boldsymbol{\Sigma}}\;\boldsymbol{X}(\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\;\boldsymbol{X}^\mathsf{T}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)\,, \tag{11.8}$$

obviously, independent on the scaling of the covariance matrices.

With the hat matrix

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\;\boldsymbol{X}^\mathsf{T} \tag{11.9}$$

this is equivalent to analysing

$$\lambda(\boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}}\tilde{\boldsymbol{\Sigma}}^{-1}_{\hat{\theta}\hat{\theta}}) = \lambda(\boldsymbol{H}\tilde{\boldsymbol{\Sigma}}\boldsymbol{H}\tilde{\boldsymbol{\Sigma}}^{-1}) \geq 1\,, \tag{11.10}$$

which is a unitless quantity. Due to the Gauss–Markov theorem his quantity always is not smaller than 1, i.e., – as to be expected – the approximate solution generally is less accurate than the optimal.

11.5 The weighted arithmetic mean

We want to investigate the effect of using a wrong covariance matrix in case of diagonal covariance matrices,

$$\tilde{\boldsymbol{\Sigma}} = \mathrm{Diag}([\widetilde{\sigma_i^2}]) \quad \text{and} \quad \boldsymbol{\Sigma} = \boldsymbol{I}\,, \tag{11.11}$$

11.5.1 The effect of using equal weights

We start with a simple example, the weighted arithmetic mean of N observations. The design matrix for the arithmetic mean is

$$X = \mathbf{1}\,. \tag{11.12}$$

Then with $X^\mathsf{T}X = \mathbf{1}^\mathsf{T}\mathbf{1} = N$ Eq. (11.10) reduces to

$$\begin{aligned}
\lambda(\Sigma_{\hat{\theta}\hat{\theta}}\tilde{\Sigma}^{-1}_{\hat{\theta}\hat{\theta}}) &= \frac{\sigma^2_{\hat{\theta}}}{\tilde{\sigma}^2_{\hat{\theta}}} && (11.13)\\
&= \frac{1}{N^2}\lambda\left(\mathbf{1}\mathbf{1}^\mathsf{T}\tilde{\Sigma}\mathbf{1}\mathbf{1}^\mathsf{T}\tilde{\Sigma}^{-1}\right) && (11.14)\\
&= \frac{1}{N^2}\lambda\left(\mathbf{1}^\mathsf{T}\tilde{\Sigma}\mathbf{1}\cdot\mathbf{1}^\mathsf{T}\tilde{\Sigma}^{-1}\mathbf{1}\right) && (11.15)\\
&= \frac{1}{N^2}\mathrm{tr}\tilde{\Sigma}\cdot\mathrm{tr}\tilde{\Sigma}^{-1} && (11.16)\\
&= \frac{\sum_{n=1}^N \tilde{\sigma}^2_n}{N}\cdot\frac{\sum_{n=1}^N w_n}{N} && (11.17)\\
&= \overline{\tilde{\sigma}^2_n}\cdot\overline{w_n} && (11.18)\\
&= \frac{\mu^{(a)}_{\tilde{\sigma}^2}}{\mu^{(h)}_{\tilde{\sigma}^2}} \geq 1\,. && (11.19)
\end{aligned}$$

or the ratio of the arithmetic mean $\mu^{(a)}_{\tilde{\sigma}^2} = \overline{\tilde{\sigma}^2_n}$ and the harmonic mean $\mu^{(h)}_{\tilde{\sigma}^2} = \left(\overline{1/\tilde{\sigma}^2_n}\right)^{-1}$ of the variances or of the weights. This ratio always is larger than 1 except all variances are identical.

11.5.2 Modeling the weights using the Gamma-distribution

The Gamma-distribution is a useful model for the weights, since it is the conjugate prior for the precision $w = 1/\sigma^2$ of the Gaussian distribution with known mean.

Let the weights be Gamma distributed

$$\underline{w}_n \sim \mathrm{Gamma}(\alpha,\beta) = \mathrm{Gamma}(k,\theta) \tag{11.20}$$

where the two parametrizations are related by

$$k = \alpha \quad \text{and} \quad \theta = \frac{1}{\beta}\,. \tag{11.21}$$

The mean and the variance are given by

$$\mathbb{E}(\underline{w}_n) = \frac{\alpha}{\beta} = k\theta \quad \text{and} \quad \mathbb{V}(\underline{w}_n) = \frac{\alpha}{\beta^2} = k\theta^2\,. \tag{11.22}$$

So, given a mean weight μ_w and a variance of the weights σ^2_w we may choose the parameters

$$\alpha = \frac{\mu^2_w}{\sigma^2_w} \quad \text{and} \quad \beta = \frac{\sigma^2_w}{\mu_w}\,. \tag{11.23}$$

The inverse weights, thus the variances follow an inverse Gamma distribution

$$\underline{\sigma}^2_n \sim \mathrm{invGamma}(a,b) \tag{11.24}$$

with the same parameters. Their mean is

$$\mathbb{E}(\underline{\sigma}^2_n) = \frac{\beta}{\alpha-1} \quad \text{for } \alpha > 1 \quad \text{and} \quad \mathbb{V}(\underline{\sigma}^2_n) = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}\,. \tag{11.25}$$

For values $\alpha \leq 1$ the inverse Gamma distribution has no finite mean, similar to the variance of the Cauchy distribution. This is plausible, since then the likelihood of small weights thus large variances is very high.

Hence the product of the means of the variances and the weights is given by

$$\lambda = \frac{\alpha}{\beta} \cdot \frac{\beta}{\alpha - 1} = \frac{\alpha}{\alpha - 1} \geq 1 \tag{11.26}$$

So, in case the weights on an average are μ_w and have a standard deviation of $\sigma_w = c \cdot \mu_w$, thus

$$c = \frac{\sigma_w}{\mu_w}\,, \tag{11.27}$$

we obtain

$$\lambda = \frac{\sigma^2_{\hat{\theta}}}{\tilde{\sigma}^2_{\hat{\theta}}} = \frac{\frac{\mu_w^2}{\sigma_w^2}}{\frac{\mu_w^2}{\sigma_w^2} - 1} = \frac{1}{1 - c^2}\,. \tag{11.28}$$

For values $c \geq 1$ the ratio of the variances is unlimited.

11.6 An example

We take as an example the mean of two points in the plane, and compare the arithmetic mean with the statistically optimal mean.

The Fig. 11.1 shows the arithmetic mean and the weighted mean (centroids) of two

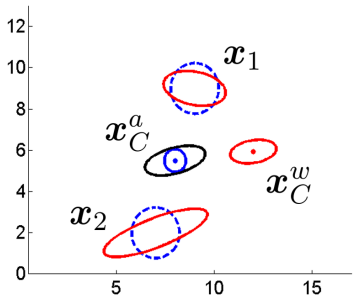


Figure 11.1: Simple mean $\boldsymbol{x}_C^a$ and weighted mean $\boldsymbol{x}_C^w$ of two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with strongly anisotropic uncertainty (red standard ellipses). The centroid determined as weighted mean clearly lies outside the line joining the two points.

points. They are assumed to be mutually independent. Their uncertainty is different and anisotropic (red standard ellipses). The centroids result from the two models

$$\begin{bmatrix} \underline{\boldsymbol{x}}_1 \\ \underline{\boldsymbol{x}}_1 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} I_2 \\ I_2 \end{bmatrix} \boldsymbol{x}_C^a, \sigma^2 I_4 \right) \tag{11.29}$$

and

$$\begin{bmatrix} \underline{\boldsymbol{x}}_1 \\ \underline{\boldsymbol{x}}_1 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} I_2 \\ I_2 \end{bmatrix} \boldsymbol{x}_C^w, \mathrm{Diag}(\{\Sigma_{11}, \Sigma_{22}\}) \right)\,. \tag{11.30}$$

The variance σ^2 in model (11.29) was assumed to be the mean of the two variances in model (11.30), s. the two blue circles.

Explicitly, the centroids are

$$\hat{\boldsymbol{x}}_C^a = \frac{1}{2}(\boldsymbol{x}_1 + \boldsymbol{x}_2) \quad \text{und} \quad \hat{\boldsymbol{x}}_C^w = (\Sigma_{11}^{-1} + \Sigma_{22}^{-1})^{-1}(\Sigma_{11}^{-1}\boldsymbol{x}_1 + \Sigma_{22}^{-1}\boldsymbol{x}_2)\,. \tag{11.31}$$

The simple arithmetic mean lies in the middle of the two points $\boldsymbol{x}_1$ und $\boldsymbol{x}_2$ on the connecting line. The weighted mean, however, lies significantly off the connecting line. The

uncertainty of the two given points allows, that the centroid may be more easily shifted in the direction of the major axes of the standard ellipses.

The standard ellipses around represent centroids are

- the covariance matrix (blue circle) of the arithmetic mean, assuming the same isotropic accuracy (blue dashed circles) of the two points. It clearly overestimates its accuracy, compared to
- the covariance matrix (black ellipse) of the arithmetic mean, assuming the anisotropic accuracy (red circles around the points), and
- the covariance matrix (red ellipse) of the weighted mean, which is smaller than the accuracy of the arithmetic mean, when assuming the known uncertainty of the point during estimation.

12 Bias of Uncertain Bilinear Forms of Stochastic Quantities

Bilinear forms of stochastic quantities lead to a bias caused by the nonlinear relation. The bias is analysed assuming a Gaussian distribution for the given variables.

12.1 Preface

This note (2021) addresses the concatenation and testing of uncertain geometric entities. They lead to quadratic forms, which, in case of Gaussian distributions lead to biases caused by the nonlinear relations. The notes collects the basic equations for these biases using moments up to 4-th order.

12.2 Motivation

Concatenation and testing of uncertain geometric entities leads to quadratic forms, which, in case of Gaussian distributions lead to biases caused by the nonlinear relation. We derive expressions for the mean and the variance of nonlinear functions of stochastical variables, which depend on higher order derivatives and moments.

On notation. We use the following abbreviations for a function $\underline{y} = f(\underline{\boldsymbol{x}})$

$$\mu_f = f(\boldsymbol{\mu}), \quad \mu_g = g(\boldsymbol{\mu}), \quad \boldsymbol{f}_x = [f_{x_i}] = \left[\frac{\partial f}{\partial x_i}\right]\Big|_{x=\mu}, \quad \boldsymbol{g}_x = [g_{x_i}] = \left[\frac{\partial g}{\partial x_i}\right]\Big|_{x=\mu}, \tag{12.1}$$

and

$$\mathsf{F}_{xx} = [f_{x_i x_j}] = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]\Big|_{x=\mu}, \quad \mathsf{G}_{xx} = [g_{x_i x_j}] = \left[\frac{\partial^2 g}{\partial x_i \partial x_j}\right]\Big|_{x=\mu}. \tag{12.2}$$

Furthermore, for a vector of normally distributed random variables with zero mean

$$\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}) \quad \text{with} \quad \boldsymbol{\Sigma} = [\sigma_{ij}] \tag{12.3}$$

we use the expectations of even order[1]

$$\begin{aligned}
\mathbb{E}(\underline{x}_i^2) &= \sigma_{ii} && (12.4)\\
\mathbb{E}(\underline{x}_i\underline{x}_j) &= \sigma_{ij} && (12.5)\\
\mathbb{E}(\underline{x}_i^4) &= 3\sigma_{ii}^2 && (12.6)\\
\mathbb{E}(\underline{x}_i^3\underline{x}_j) &= 3\sigma_{ii}\sigma_{ij} && (12.7)\\
\mathbb{E}(\underline{x}_i^2\underline{x}_j^2) &= \sigma_{ii}\sigma_{jj} + 2\sigma_{ij} && (12.8)\\
\mathbb{E}(\underline{x}_i^2\underline{x}_j\underline{x}_k) &= \sigma_{ii}\sigma_{jk} + 2\sigma_{ij}\sigma_{ik} && (12.9)\\
\mathbb{E}(\underline{x}_i\underline{x}_j\underline{x}_k\underline{x}_l) &= \sigma_{ij}\sigma_{kl} + \sigma_{ik}\sigma_{jl} + \sigma_{il}\sigma_{jk} && (12.10)
\end{aligned}$$

The expectations of odd order are zero.

12.3 A scalar function of a stochastical vector

For a scalar function

$$\underline{y} = f(\underline{\boldsymbol{x}}) \quad \text{with} \quad \underline{\boldsymbol{x}} \sim \mathcal{M}(\boldsymbol{\mu}_x, \Sigma_{xx}) \tag{12.11}$$

of a vector valued stochastical variable $\underline{\boldsymbol{x}}$ the mean is given by, see Förstner and Wrobel (2016, Eq. (2.151))

$$\boxed{\mu_y = f(\boldsymbol{\mu}_x) + \tfrac{1}{2}\mathrm{tr}(H_{xx}\Sigma_{xx}) + O(f^n, m_n) \quad \text{with} \quad n > 2} \tag{12.12}$$

with the Hessian $H_{xx} = \partial^2 f(\boldsymbol{x})/(\partial \boldsymbol{x})^2$ up to order $n = 2$, the remaining parts depending on the higher order derivatives and the higher moments of the distribution.

Proof: We have the Taylor expansion at the mean using

$$\underline{\boldsymbol{e}} = \underline{\boldsymbol{x}} - \boldsymbol{\mu}_x \tag{12.13}$$

up to fourth order

$$\underline{y} = f(\boldsymbol{\mu}_x) + \sum_i f_{x_i}\underline{e}_i + \frac{1}{2}\sum_{ij} f_{x_i x_j}\underline{e}_i\underline{e}_j + \frac{1}{6}\sum_{ijk} f_{x_i x_j x_k}\underline{e}_i\underline{e}_j\underline{e}_k + \frac{1}{24}\sum_{ijkl} f_{x_i x_j x_k x_l}\underline{e}_i\underline{e}_j\underline{e}_k\underline{e}_l + O(f_n, m_n) \quad \text{with} \quad n > 4 \tag{12.14}$$

which depends on the derivatives of f up to order 4. Taking expectation, we obtain the mean value

$$\begin{aligned}
\mu_y &= \mu_f + \boldsymbol{f}_x^\mathsf{T}(\mathbb{E}(\underline{\boldsymbol{e}}) + \frac{1}{2}\mathbb{E}\left[\underline{\boldsymbol{e}}^\mathsf{T} F_{xx}\underline{\boldsymbol{e}}\right] && (12.15)\\
&= \mu_f + 0 + \frac{1}{2}\mathbb{E}(\mathrm{tr}(F_{xx}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T})) && (12.16)\\
&= \mu_f + \frac{1}{2}\mathrm{tr}(F_{xx}\Sigma_{xx}) && (12.17)
\end{aligned}$$

Hence, we have the mean up to first and second order approximation

$$\begin{aligned}
\mu_y^{(1)} &= f(\boldsymbol{\mu}_x)\,, && (12.18)\\
\mu_y^{(2)} &= \mu_y^{(1)} + \frac{1}{2}\mathrm{tr}(F_x x\Sigma_{xx})\,, && (12.19)
\end{aligned}$$

◇

Similarly we find the varaince of the variable $\underline{y}$

$$\boxed{\sigma_y^2 = \boldsymbol{f}_x^\mathsf{T}\Sigma_{xx}\boldsymbol{f}_x + \tfrac{1}{4}\mathrm{tr}^2(F_{xx}\Sigma_{xx}) + 2\mathrm{tr}(F_{xx}\Sigma_{xx}F_{xx}\Sigma_{xx}) + O(f_n, m_n) \quad \text{with} \quad n > 4} \tag{12.20}$$

up to order 4.

[1] see https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Proof: The covariance matrix of a random vector generally is

$$\mathbb{V}(\underline{\boldsymbol{x}}) = \mathbb{E}((\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))(\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))^{\mathsf{T}}) = \mathbb{E}(\underline{\boldsymbol{x}}\underline{\boldsymbol{x}}^{\mathsf{T}}) - \mathbb{E}(\underline{\boldsymbol{x}})\mathbb{E}(\underline{\boldsymbol{x}}^{\mathsf{T}}) \tag{12.21}$$

We first determine

$$\begin{aligned} [\mathbb{E}(\underline{y})]^2 &= (\mu_f + \frac{1}{2}\mathrm{tr}(\mathsf{F}_x x \boldsymbol{\Sigma}_{xx}))^2 && (12.22)\\ &= \mu_f^2 + \mu_f \mathrm{tr}(\mathsf{F}_x x \boldsymbol{\Sigma}_{xx}) + \frac{1}{4}\mathrm{tr}^2(\mathsf{F}_x x \boldsymbol{\Sigma}_{xx})\,. && (12.23)\end{aligned}$$

Then we take the square of $\underline{y}$

$$\begin{aligned} \underline{y}^2 &= (\mu_f + \boldsymbol{f}_x^{\mathsf{T}}(\underline{\boldsymbol{x}} - \mu_x) + \frac{1}{2}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^{\mathsf{T}}\mathsf{F}_{xx}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)) && (12.24)\\ &= \mu_f^2 + 2\mu_f \boldsymbol{f}_x^{\mathsf{T}}(\underline{\boldsymbol{x}} - \mu_x) + \frac{1}{2}\mu_f(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^{\mathsf{T}}\mathsf{F}_{xx}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)) && (12.25)\\ &\quad + \boldsymbol{f}_x^{\mathsf{T}}(\underline{\boldsymbol{x}} - \mu_x)(\underline{\boldsymbol{x}} - \mu_x)^{\mathsf{T}}\boldsymbol{f}_x + \frac{1}{2}\boldsymbol{f}_x^{\mathsf{T}}(\underline{\boldsymbol{x}} - \mu_x)(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^{\mathsf{T}}\mathsf{F}_{xx}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)) && (12.26)\\ &\quad + \frac{1}{4}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^{\mathsf{T}}\mathsf{F}_{xx}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x))(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x)^{\mathsf{T}}\mathsf{F}_{xx}(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x))\,. && (12.27)\end{aligned}$$

We now need moments of the deviations $\underline{\boldsymbol{e}}_x = \underline{\boldsymbol{x}} - \boldsymbol{\mu}_x$ third and fourth order, which depend on the distribution of $\underline{\boldsymbol{x}}$. We assume the distribution has the same moments up to fourth order as the normal distribution, sometimes called *quasi-normally distributed*. Then all odd moments are zero, and we have the relations for the scalar functions

$$\mathbb{E}(\underline{\boldsymbol{e}}_x^{\mathsf{T}}\mathsf{A}\underline{\boldsymbol{e}}_x) = \mathrm{tr}(\mathsf{A}\boldsymbol{\Sigma}_{xx}) \quad \text{and} \quad \mathbb{E}(\underline{\boldsymbol{e}}^{\mathsf{T}}\mathsf{A}\underline{\boldsymbol{e}}_x\underline{\boldsymbol{e}}_x^{\mathsf{T}}\mathsf{B}\underline{\boldsymbol{e}}_x) = \mathrm{tr}(\mathsf{A}\boldsymbol{\Sigma}_{xx})\mathrm{tr}(\mathsf{B}\boldsymbol{\Sigma}_{xx}) + 2\mathrm{tr}(\mathsf{A}\boldsymbol{\Sigma}_{xx}\mathsf{B}\boldsymbol{\Sigma}_{xx})\,. \tag{12.28}$$

Based on this assumption and the given relations we obtain the expectation of $\underline{y}^2$

$$\begin{aligned} \mathbb{E}(\underline{y}^2) &= \mu_f^2 + 0 + \frac{1}{2}\mu_f \mathrm{tr}(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx}) && (12.29)\\ &\quad + \boldsymbol{f}_x^{\mathsf{T}}\boldsymbol{\Sigma}_{xx}^{\mathsf{T}}\boldsymbol{f}_x + 0 && (12.30)\\ &\quad + \frac{1}{4}\mathrm{tr}(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx})\mathrm{tr}(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx}) + 2\mathrm{tr}(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx}\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx})\,. && (12.31)\end{aligned}$$

Hence, we obtain for the variance of $\underline{y}$ the two approximations

$$\begin{aligned} [\sigma_y^2]^{(2)} &= \boldsymbol{f}_x^{\mathsf{T}}\boldsymbol{\Sigma}_{xx}\boldsymbol{f}_x\,. && (12.32)\\ [\sigma_y^2]^{(4)} &= [\sigma_y^2]^{(2)} + \frac{1}{4}\mathrm{tr}^2(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx}) + 2\mathrm{tr}(\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx}\mathsf{F}_{xx}\boldsymbol{\Sigma}_{xx})\,. && (12.33)\end{aligned}$$

◇

12.4 Problem

Constructing geometric entities using homogenous coordinates leads to bilinear forms. Examples are the generation of lines from two points in 2D and 3D:

$$\mathbf{l} = \mathbf{x} \times \mathbf{y} \qquad \mathbf{L} = \begin{bmatrix} Y_h\boldsymbol{X}_0 - X_h\boldsymbol{Y}_0 \\ \boldsymbol{X}_0 \times \boldsymbol{Y}_0 \end{bmatrix}$$

Generally, we have the form

$$\mathbf{c} = \mathsf{A}(\mathbf{a})\mathbf{b} = \mathsf{B}(\mathbf{b})\mathbf{a}$$

where $\mathsf{A}(\mathbf{a})$ and $\mathsf{B}(\mathbf{b})$ are homogeneous matrices with entries which are linear in the vectors $\mathbf{a}$ and $\mathbf{b}$.

In case the entities are treated as stochastical variables, thus as vectors with a covariance matrix, error propagation appears to be simple, as the two matrices are the Jacobians:

$$\frac{\partial \mathbf{c}}{\partial \mathbf{b}} = \mathsf{A}(\mathbf{a}) \qquad \frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \mathsf{B}(\mathbf{b})$$

yielding

$$\Sigma_{cc} = (\mathsf{B}(\mathbf{b}), \mathsf{A}(\mathbf{a})) \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \begin{bmatrix} \mathsf{B}^\mathsf{T}(\mathbf{b}) \\ \mathsf{A}^\mathsf{T}(\mathbf{a}) \end{bmatrix}$$

or in case of independence

$$\Sigma_{cc} = \mathsf{B}(\mathbf{b})\Sigma_{aa}\mathsf{B}^\mathsf{T}(\mathbf{b}) + \mathsf{A}(\mathbf{a})\Sigma_{bb}\mathsf{A}^\mathsf{T}(\mathbf{a})$$

However, though the forms are linear in each variable, the form is nonlinear, as products of variables appear. Thus, error propagation is not rigorous, in the sense, that the first two moments are not sufficient to describe the distribution of the product.

The goal of this note is to investigate the bias which has to be expected. We especially investigate the bias of the most simple bilinear relations and

$$z = xy \quad \text{and} \quad z = f(\boldsymbol{x})g(\boldsymbol{x})\,.$$

12.5 Rigorous Mean and Variance for $\underline{z} = \underline{x}\underline{y}$

We start from the expansion, actually the Taylor series at the mean values, which in this case is finite:

$$\underline{z} = \mu_z + \underline{e}_z = (\mu_x + \underline{e}_x)(\mu_y + \underline{e}_y) = \mu_x\mu_y + \mu_y\underline{e}_x + \mu_x\underline{e}_y + \underline{e}_x\underline{e}_y \tag{12.34}$$

We assume to have some arbitrary distribution M

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}\right) = \mathcal{M}(\boldsymbol{\mu}, \Sigma) \tag{12.35}$$

with mean $\boldsymbol{\mu}$ and covariance matrix Σ.

Taking the expectation, we obtain the rigorous expression:

$$\mathbb{E}(\underline{z}) = \mathbb{E}(\mu_x\mu_y + \mu_y\underline{e}_x + \mu_x\underline{e}_y + \underline{e}_x\underline{e}_y) = \mu_x\mu_y + \sigma_{xy} \tag{12.36}$$

The variance can be computed from

$$\mathbb{V}(\underline{z}) = \mathbb{E}([\underline{z} - \mathbb{E}(\underline{z})]^2) = \mathbb{E}(\underline{z}^2) - [\mathbb{E}(\underline{z})]^2 \tag{12.37}$$

We immediately get

$$[\mathbb{E}(\underline{z})]^2 = \mu_x^2\mu_y^2 + 2\mu_x\mu_y\sigma_{xy} + \sigma_{xy}^2 \tag{12.38}$$

We now have

$$\begin{aligned} \underline{z}^2 &= (\mu_x\mu_y + \mu_y\underline{e}_x + \mu_x\underline{e}_y + \underline{e}_x\underline{e}_y)^2 && (12.39)\\ &= \underline{\mu_x^2\mu_y^2} + \underline{\mu_y^2\underline{e}_x^2} + \underline{\mu_x^2\underline{e}_y^2} + \underline{\underline{e}_x^2\underline{e}_y^2} + && (12.40)\\ &+ 2(\mu_x\mu_y^2\underline{e}_x + \mu_x^2\mu_y\underline{e}_y + \underline{\mu_x\mu_y\underline{e}_x\underline{e}_y} + && (12.41)\\ &+ \underline{\mu_y\underline{e}_x\mu_x\underline{e}_y} + \mu_y\underline{e}_x^2\underline{e}_y + \mu_x\underline{e}_y^2\underline{e}_x) && (12.42) \end{aligned}$$

The forth moments depend on the type of distribution. For normally distributed variables we have

$$\mathbb{E}(\underline{e}_x^2\underline{e}_y^2) = \int\int (x-\mu_x)^2(y-\mu_y)^2 p_{xy}(x,y)dx\,dy = \sigma_x^2\sigma_y^2 + 2\sigma_{xy}^2$$

Therefore we obtain the expected value

$$\mathbb{E}(\underline{z}^2) = \mu_x^2\mu_y^2 + \mu_y^2\sigma_x^2 + \mu_x^2\sigma_y^2 + \sigma_x^2\sigma_y^2 + 2\sigma_{xy}^2 + 4\mu_x\mu_y\sigma_{xy} \tag{12.43}$$

This yields the variance of $\underline{z}$

$$\sigma_z^2 = \mu_y^2\sigma_x^2 + \mu_x^2\sigma_y^2 + 2\mu_x\mu_y\sigma_{xy} + \sigma_x^2\sigma_y^2 + \sigma_{xy}^2 \tag{12.44}$$

12.6 Linear Approximation of Mean and Variance of $\underline{z} = \underline{x}\underline{y}$

The mean in a first order approximation is

$$\mu_z^{(1)} = \mu_x \mu_y \tag{12.45}$$

With the Jacobian

$$J = (y, x)$$

Using classical error propagation, we obtain the variance

$$\sigma_z^{2(1)} = J \Sigma J^{\mathsf{T}} = \mu_x^2 \mu_y^2 + \mu_y^2 \sigma_x^2 + 2\mu_x \mu_y \sigma_{xy} \tag{12.46}$$

12.7 Bias of Linear Approximation of Mean and Variance

The bias in mean is

$$b_{\mu_z} = \mu_z^{(1)} - \mu_z = -\sigma_{xy} \tag{12.47}$$

It is zero if the two variables are uncorrelated.

The bias in variance is

$$b_{\sigma_z^2} = \sigma_z^{2(1)} - \sigma_z^2 = -\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2 = -\sigma_x^2 \sigma_y^2 (1 + \rho_{xy}^2) \tag{12.48}$$

It is *not* zero for uncorrelated variables. Actually, the variance is *underestimated* if one relies on classical error propagation, as $\sigma_z^{2(1)} < \sigma_z^2$ for uncorrelated variables, see Haddon and Forsyth (2001).

The relative bias of the variance, i. e. the bias related to the variance is

$$r_{\sigma_z^2} = \frac{b_{\sigma_z^2}}{\sigma_z^2} = \frac{-\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2}{\mu_y^2 \sigma_x^2 + \mu_x^2 \sigma_y^2 + 2\mu_x \mu_y \sigma_{xy} + \sigma_x^2 \sigma_y^2 + \sigma_{xy}^2} \tag{12.49}$$

In order to get an impression on the size we assume $\sigma_x = \sigma_y = \sigma$ and $\sigma_{xy} = 0$ and obtain

$$r_{\sigma_z^2} = -\frac{\sigma^2}{\mu_x^2 + \mu_y^2 + \sigma^2} \tag{12.50}$$

Thus only in case $\mu_x^2 + \mu_y^2 < \sigma^2$ the relative bias in variance is larger than 50 % of the variance. These cases have been discussed by Hannon/Forsyth.

12.8 Probability Density Function for Zero Mean Variables

In case of independent zero mean Gaussian variables

$$\underline{x} \sim N(0, \sigma^2) \qquad \underline{y} \sim N(0, \sigma^2)$$

we find the probability density function of $\underline{z}$ from

$$p_z(z) = \int_0^\infty \frac{2}{u} p_x(u) p_y\left(\frac{z}{u}\right) du$$

yielding

$$p_z(z) = \frac{\text{BesselK}\left(0, \frac{|z|}{\sigma^2}\right)}{\pi \sigma^2}$$

with the Bessel function BesselK(x) of the second kind. It definitely is not normally distributed (cf. fig. 12.1), but has the variance

$$\mathbb{V}(\underset{\sim}{z}) = 2\int_{z=0}^{\infty} z^2 p_z(z)dz = \sigma^4$$

in accordance with (12.44).

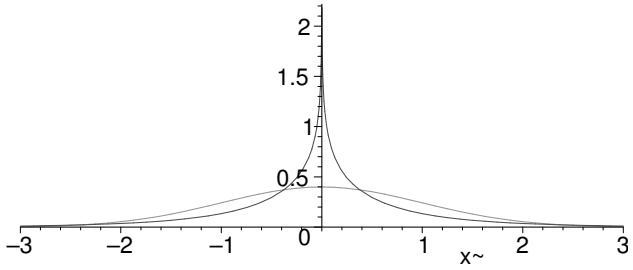


Figure 12.1: The Bessel function is the pdf of the product of two zero-mean Gaussian variables. Shown is the Bessel function and the Gaussian with the same variance

12.9 Generalization to the Cross Product

We now generalize the result to general bilinear forms. We take as an example the previous one $z = xy$ and

$$\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b} = S(\boldsymbol{a})\boldsymbol{b} = -S(\boldsymbol{b})\boldsymbol{a}$$

We assume the general bilinear form

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}) = B(\boldsymbol{b})\boldsymbol{a} = A(\boldsymbol{a})\boldsymbol{b} = U(\boldsymbol{x})\boldsymbol{x}$$

with

$$U(\boldsymbol{x}) = \frac{1}{2}[B(\boldsymbol{b}), A(\boldsymbol{a})]$$

and the vector

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$$

or in components

$$y_i = f_i(\boldsymbol{x}) = f_i(\boldsymbol{a}, \boldsymbol{b}) = \sum_j B_{ij}(b_k)a_j = \sum_k A_{ik}(a_j)b_k = \sum_l U_{il}(x_m)x_l = \boldsymbol{u}_i^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{x}$$

We have the Hessians (indexed i)

$$W_i = (W_{ilm}) = \left(\frac{\partial y_i^2}{\partial x_l \partial x_m}\right) = \left(\frac{\partial U_{il}(x_m)}{\partial x_m}\right)$$

which only contains values -1, 0 or 1, as the B_{ij} and the A_{ik} and thus the U_{il} are linear in the variables. Actuall we may partition W_i into

$$W_i = \begin{bmatrix} \mathbf{0} & W_i^{(b)} \\ W_i^{(a)} & \mathbf{0} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \mathbf{0} & \dfrac{\partial \boldsymbol{B}_i(\boldsymbol{b})}{\partial \boldsymbol{b}} \\ \dfrac{\partial \boldsymbol{A}_i(\boldsymbol{a})}{\partial \boldsymbol{a}} & \mathbf{0} \end{bmatrix}$$

We therefore may write the components of the bilinear form as quadratic forms

$$y_i = \boldsymbol{x}^{\mathsf{T}} W_i \boldsymbol{x}$$

We now easily can give the expected value (cf. Koch (1999, Sect. 27, (2.174)))

$$\mathbb{E}(\underline{y}_i) = \mathbb{E}(\underline{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{W}_i \underline{\boldsymbol{x}}) = \boldsymbol{\mu}_x^{\mathsf{T}} \boldsymbol{W}_i \boldsymbol{\mu}_x + \mathrm{tr} \boldsymbol{W}_i \boldsymbol{\Sigma}$$

which can be simplified to

$$\mathbb{E}(\underline{y}_i) = f_i(\boldsymbol{\mu}_x) + \mathrm{tr} \boldsymbol{W}_i \boldsymbol{\Sigma} = f_i(\boldsymbol{\mu}_x) + \mathrm{tr}(\boldsymbol{W}_i^{(a)\mathsf{T}} + \boldsymbol{W}_i^{(b)}) \boldsymbol{\Sigma}_{ab}$$

Example 1: The observational vector is

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

The matrices (B_{ij}) and (A_{ik}) are

$$B = y \qquad A = x$$

The matrix $\boldsymbol{U} = (U_{il})$ therefore is

$$\boldsymbol{U} = \frac{1}{2}[y,\ x]$$

leading to the function

$$\mathbf{z} = \frac{1}{2}[y,\ x] \begin{bmatrix} x \\ y \end{bmatrix}$$

In our case we e. g. obtain the single Hessian

$$\boldsymbol{W}_1 = (W_{1lm}) = \frac{1}{2} \begin{bmatrix} \mathbf{0} & 1 \\ 1 & 0 \end{bmatrix}$$

This we have the mean

$$\mathbb{E}(\underline{z}_i) = (\mu_x \mu_b) + \sigma_{ab}$$

as to be expected.
In case of uncorrelated vectors $\mathbf{x}$ and $\mathbf{y}$ we obtain

$$\mathbb{E}(\underline{\mathbf{z}}_i) = \mu_x \boldsymbol{\mu}_y$$

$\diamond$

Example 2: The observational vector is

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

The matrices (B_{ij}) and (A_{ik}) are

$$-\mathsf{S}(\mathbf{y}) \qquad \mathsf{S}(\mathbf{x})$$

The matrix $U = (U_{il})$ therefore is

$$U = \frac{1}{2}[-\mathsf{S}(\mathbf{y}),\ \mathsf{S}(\mathbf{x})] = \frac{1}{2}\left(\begin{array}{ccc|ccc} 0 & y_3 & -y_2 & 0 & -x_3 & x_2 \\ -y_3 & 0 & y_1 & x_3 & 0 & -x_1 \\ y_2 & -y_1 & 0 & -x_2 & x_1 & 0 \end{array}\right)$$

leading to the function

$$\mathbf{l} = \frac{1}{2}[-\mathsf{S}(\mathbf{y}),\ \mathsf{S}(\mathbf{x})] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

In our case we e. g. obtain the Hessian

$$W_1 = (W_{1lm}) = \frac{1}{2}\begin{bmatrix} \mathbf{0} & -S_{e_1} \\ S_{e_1} & \mathbf{0} \end{bmatrix} = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)$$

generally

$$W_i = (W_{ilm}) = \frac{1}{2}\begin{bmatrix} \mathbf{0} & -S_{e_i} \\ S_{e_i} & \mathbf{0} \end{bmatrix}$$

Thus, we have the mean

$$\mathbb{E}(\underline{\mathbf{l}}_i) = (\boldsymbol{\mu}_a \times \boldsymbol{\mu}_b)_i + \operatorname{trace}(\mathsf{S}_{e_i}\boldsymbol{\Sigma}_{ab})$$

In case of uncorrelated vectors $\mathbf{x}$ and $\mathbf{y}$ we obtain

$$\mathbb{E}(\underline{\mathbf{l}}_i) = (\boldsymbol{\mu}_x \times \boldsymbol{\mu}_y)_i + (-1)^{i+1}\sigma_{x_{i+1}y_{i+2}} + (-1)^i\sigma_{y_{i+1}x_{i+2}}$$

where the indices have to be taken cyclically. In case of isotropic errors, i. e. no correlation between the components of $\mathbf{x}$ and $\mathbf{y}$ the bias is zero. $\diamond$

The variance depends on the moments up to 4th order, which in case of Gaussian variables only depend on the first two moments. We obtain the covariance of two different entries of $\underline{\boldsymbol{y}}$ (cf. Koch (1999, Sect. 27, (2.175)))

$$\mathbb{C}\mathrm{ov}(\underline{y}_i\underline{y}_j) = \mathbb{C}\mathrm{ov}(\underline{\boldsymbol{x}}^{\mathsf{T}}W_i\underline{\boldsymbol{x}}, \underline{\boldsymbol{x}}^{\mathsf{T}}W_j\underline{\boldsymbol{x}}) = 2\mathrm{tr}W_i\boldsymbol{\Sigma}W_j\boldsymbol{\Sigma} + 4\boldsymbol{\mu}_x^{\mathsf{T}}W_i\boldsymbol{\Sigma}W_j\boldsymbol{\mu}_x$$

which specialized to the variance

$$\mathbb{V}(\underline{y}_i) = 2\mathrm{tr}W_i\boldsymbol{\Sigma}W_i\boldsymbol{\Sigma} + 4\boldsymbol{\mu}_x^{\mathsf{T}}W_i\boldsymbol{\Sigma}W_i\boldsymbol{\mu}_x$$

Due to the bilinearity this expression can further be simplified. We obtain

$$\begin{aligned}
\sigma^2_{y_i} &= 2\mathrm{tr}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix} \\
&+ 4[\boldsymbol{\mu}_b W_i^{(a)},\ \boldsymbol{\mu}_a W_i^{(a)}]\Sigma\begin{bmatrix} W_i^{(b)}\boldsymbol{\mu}_a \\ W_i^{(a)}\boldsymbol{\mu}_b \end{bmatrix} \\
&= 2\mathrm{tr}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix} \\
&+ [B_i^{\mathsf{T}}(\boldsymbol{b}),\ A_i^{\mathsf{T}}(\boldsymbol{a})]\Sigma\begin{bmatrix} B_i(\boldsymbol{b}) \\ A_i(\boldsymbol{a}) \end{bmatrix}
\end{aligned}$$

the second term obviously being the result of classical error propagation. The bias is

$$b_{\sigma^2_{y_i}} = \mathrm{tr}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix}\begin{bmatrix} W_i^{(b)}\Sigma_{ba} & W_i^{(b)}\Sigma_{bb} \\ W_i^{(a)}\Sigma_{aa} & W_i^{(a)}\Sigma_{ab} \end{bmatrix}$$

Example 1: Here we have

$$W = \frac{1}{2}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

and

$$W^{(a)} = W^{(b)} = \frac{1}{2}$$

Thus the bias of the varince σ_z^2 is

$$b_{\sigma_z^2} = \sigma_x^2\sigma_y^2 + \sigma_{xy}^2$$

$\diamond$

Example 2: Here we have

$$W^{(a)} = -W^{(b)} = \frac{1}{2}S_{e_i}$$

The bias of the variance σ_z^2 is more involving. In case of uncorrelated $\boldsymbol{a}$ and $\boldsymbol{b}$ and

$$\Sigma_{aa} = \Sigma_{bb} = \sigma^2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

we have the bias in the three variances

$$b_{\sigma^2_{y_1}} = b_{\sigma^2_{y_2}} = 0 \qquad b_{\sigma^2_{y_3}} = 4\sigma^4$$

$\diamond$

12.10 Generalization to the Product of two Functions $z = f(\boldsymbol{x})g(\boldsymbol{x})$

We assume the stochastical N vector

$$\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)\,, \tag{12.51}$$

and two functions

$$f = f(\boldsymbol{x}) \quad \text{and} \quad g = g(\boldsymbol{x})\,. \tag{12.52}$$

We need to specify the distribution since we need fourth moments. We assume the distribution has the same fourth moments as the normal distribution.

The task is to determine the bias of the mean and the variance of

$$\underline{z} = f(\underline{\boldsymbol{x}})g(\underline{\boldsymbol{x}})\,. \tag{12.53}$$

We start from the Taylor expansion

$$\begin{aligned}
\underline{z} &= \mu_z + \underline{e}_z && (12.54)\\
&= f(\boldsymbol{\mu}+\underline{\boldsymbol{e}})g(\boldsymbol{\mu}+\underline{\boldsymbol{e}}) && (12.55)\\
&= \left(\mu_f + \boldsymbol{f}_x^{\mathsf{T}}\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^{\mathsf{T}}F\underline{\boldsymbol{e}}\right)\left(\mu_g + \boldsymbol{g}_x^{\mathsf{T}}\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^{\mathsf{T}}G\underline{\boldsymbol{e}}\right) && (12.56)\\
&= \quad \mu_f\mu_g && (12.57)\\
&\quad +\mu_f\boldsymbol{g}_x^{\mathsf{T}}\underline{\boldsymbol{e}} + \boldsymbol{f}_x^{\mathsf{T}}\mu_g\underline{\boldsymbol{e}} && (12.58)\\
&\quad +\frac{1}{2}\mu_f\underline{\boldsymbol{e}}^{\mathsf{T}}G\underline{\boldsymbol{e}} + \boldsymbol{f}_x^{\mathsf{T}}\underline{\boldsymbol{e}}\boldsymbol{g}_x^{\mathsf{T}}\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^{\mathsf{T}}F\underline{\boldsymbol{e}}\mu_g && (12.59)\\
&\quad +\frac{1}{2}\boldsymbol{f}_x^{\mathsf{T}}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^{\mathsf{T}}G\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^{\mathsf{T}}F\underline{\boldsymbol{e}}\boldsymbol{g}_x^{\mathsf{T}}\underline{\boldsymbol{e}} && (12.60)\\
&\quad +\frac{1}{4}\underline{\boldsymbol{e}}^{\mathsf{T}}F\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^{\mathsf{T}}G\underline{\boldsymbol{e}}\,. && (12.61)
\end{aligned}$$

12.10.1 The mean

We obtain – omitting the argument $\boldsymbol{\mu}_x$

$$\begin{aligned}
\mathbb{E}(\underline{z}) &= \quad \mu_f\,\mu_g && (12.62)\\
&\quad +\frac{1}{2}\mu_f\,\mathrm{tr}(G\Sigma) + \boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x + \frac{1}{2}\mu_g\,\mathrm{tr}(F\Sigma) && (12.63)\\
&\quad +\frac{1}{4}\left(\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma) + 2\mathrm{tr}(F\Sigma G\Sigma)\right)\,. && (12.64)
\end{aligned}$$

Hence, we obtain the mean in three approximations

$$\begin{aligned}
\mu_z^{(0)} &= \mu_f\,\mu_g && (12.65)\\
\mu_z^{(2)} &= \mu_z^{(0)} + \frac{1}{2}\mu_f\,\mathrm{tr}(G\Sigma) + \boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x + \frac{1}{2}\mu_g\,\mathrm{tr}(F\Sigma) && (12.66)\\
\mu_z^{(4)} &= \mu_z^{(2)} + \mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma) + 2\mathrm{tr}(F\Sigma G\Sigma)\,, && (12.67)
\end{aligned}$$

where all scalars, vectors and matrices are to be evaluated at the mean value. For $\Sigma = \sigma^2 I$ we obtain

$$\begin{aligned}
\mu_z^{(0)} &= \mu_f\,\mu_g && (12.68)\\
\mu_z^{(2)} &= \mu_z^{(0)} + \left(\frac{1}{2}f\,\mathrm{tr}(G) + \boldsymbol{g}_x^{\mathsf{T}}\boldsymbol{f}_x + \frac{1}{2}g\,\mathrm{tr}(F)\right)\sigma^2 && (12.69)\\
\mu_z^{(4)} &= \mu_z^{(2)} + \left(\mathrm{tr}(F)\mathrm{tr}(G) + 2\mathrm{tr}(FG)\right)\sigma^4\,, && (12.70)
\end{aligned}$$

Example 12.10.25: $z = xy$. If we specialize the function of a 2-vector

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} \sim \mathcal{M}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}\right) \tag{12.71}$$

to

$$f = x \quad \text{and} \quad g = y \tag{12.72}$$

we obtain the derivatives

$$\boldsymbol{f}_x = \boldsymbol{e}_1\,, \quad \boldsymbol{g}_x = \boldsymbol{e}_2, \quad \text{and} \quad F = G = 0\,. \tag{12.73}$$

The mean values then are, using from (12.65), ff.,

$$\begin{aligned}
\mu_z^{(0)} &= \mu_x\,\mu_y && (12.74)\\
\mu_z^{(2)} &= \mu_z^{(0)} + \sigma_{xy} && (12.75)\\
\mu_z^{(4)} &= \mu_z^{(2)}\,, && (12.76)
\end{aligned}$$

consistent with (12.36). ◇

12.10.2 The variance

We again use (12.37)

$$\mathbb{V}(\underline{z}) = \mathbb{E}([\underline{z} - \mathbb{E}(\underline{z})]^2) = \mathbb{E}(\underline{z}^2) - [\mathbb{E}(\underline{z})]^2 \tag{12.77}$$

We directly obtain from (12.62) up to fourth order terms in σ omitting odd terms which are zero

$$\begin{aligned}
[\mathbb{E}(\underline{z})]^2 &= (\mu_f\mu_g && (12.78)\\
&+\frac{1}{2}\mu_f \operatorname{tr}(G\Sigma) + \boldsymbol{g}_x^\mathsf{T}\Sigma\boldsymbol{f}_x + \frac{1}{2}\mu_g \operatorname{tr}(F\Sigma) && (12.79)\\
&+\frac{1}{4}\left(\operatorname{tr}(F\Sigma)\operatorname{tr}(G\Sigma) + 2\operatorname{tr}(F\Sigma G\Sigma)\right)^2 && (12.80)\\
&= \mu_f^2\mu_g^2 && (12.81)\\
&+\left[\mu_f^2\mu_g\operatorname{tr}(G\Sigma) + 2\mu_f\mu_g\boldsymbol{g}_x^\mathsf{T}\Sigma\boldsymbol{f}_x + \mu_f\mu_g^2 \operatorname{tr}(F\Sigma)\right] && (12.82)\\
&+\Bigg[\frac{1}{2}\mu_f\mu_g\operatorname{tr}(F\Sigma)\operatorname{tr}(G\Sigma) + \mu_f\mu_g\operatorname{tr}(F\Sigma G\Sigma) && (12.83)\\
&+\frac{1}{4}\mu_f^2\operatorname{tr}^2(G\Sigma) + \mu_f \operatorname{tr}(G\Sigma)\boldsymbol{g}_x^\mathsf{T}\Sigma\boldsymbol{f}_x + \frac{1}{4}\mu_f\mu_g\operatorname{tr}(G\Sigma)\operatorname{tr}(F\Sigma) && (12.84)\\
&+(\boldsymbol{g}_x^\mathsf{T}\Sigma\boldsymbol{f}_x)^2 + \mu_g\boldsymbol{g}_x^\mathsf{T}\Sigma\boldsymbol{f}_x \operatorname{tr}(F\Sigma) + \frac{1}{4}\mu_g^2\operatorname{tr}^2(F\Sigma)\Bigg] && (12.85)
\end{aligned}$$

First, we have up to fourth order terms, omitting the odd terms of $\underline{\boldsymbol{e}}$

$$\begin{aligned}
\underline{z}^2 &= \Big(\mu_f\mu_g + \mu_f\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} + \mu_g\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}} && (12.86)\\
&+\frac{1}{2}\mu_f\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}} + \boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}}\mu_g && (12.87)\\
&+\frac{1}{2}\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}} + \frac{1}{2}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}}\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} && (12.88)\\
&+\frac{1}{4}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}}\Big)^2 && (12.89)\\
&= \mu_f^2\mu_g^2 && (12.90)\\
&+\left[\mu_f^2\mu_g\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}} + 2\mu_f\mu_g\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} + \mu_f\mu_g^2\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}}\right. && (12.91)\\
&+\left.\mu_f^2\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{g}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} + 2\mu_f\mu_g\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{g}_x\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}} + \mu_g^2\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\right] && (12.92)\\
&+\Big[\mu_f\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{g}_x\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}} + \mu_f\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{g}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}} && (12.93)\\
&+\mu_g\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}} + \mu_g\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}} && (12.94)\\
&+\frac{1}{4}\mu_f^2(\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}})^2 + \mu_f\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}}\boldsymbol{f}_x^\mathsf{T}\underline{\boldsymbol{e}}\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}} + \frac{1}{2}\mu_f\mu_g\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}} && (12.95)\\
&+(\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}})^2 + \mu_g\underline{\boldsymbol{e}}^\mathsf{T}\boldsymbol{f}_x\boldsymbol{g}_x^\mathsf{T}\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}} + \frac{1}{4}\mu_g^2(\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}})^2 && (12.96)\\
&+\frac{1}{16}(\underline{\boldsymbol{e}}^\mathsf{T}F\underline{\boldsymbol{e}}\underline{\boldsymbol{e}}^\mathsf{T}G\underline{\boldsymbol{e}})^2\Big] && (12.97)
\end{aligned}$$

The expectation is

$$
\begin{aligned}
\mathbb{E}(\underline{z}^2) &= \mu_f^2\mu_g^2 && (12.98)\\
&+\Big[\mu_f^2\mu_g \mathrm{tr}(G\Sigma) + 2\mu_f\mu_g \boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + \mu_f\mu_g^2\mathrm{tr}(F\Sigma) && (12.99)\\
&+ \mu_f^2\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + 2\mu_f\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + +\mu_g^2\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\Big] && (12.100)\\
&+\Big[\mu_f\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(G\Sigma) + \mu_f\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(F\Sigma) && (12.101)\\
&+\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\ \mathrm{tr}(G\Sigma) + \mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(F\Sigma) && (12.102)\\
&+\frac{1}{4}\mu_f^2(\mathrm{tr}(G\Sigma))^2 + \mu_f\mathrm{tr}(G\Sigma)\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + \frac{1}{2}\mu_f\mu_g\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma) && (12.103)\\
&+(\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x)^2 + \mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\mathrm{tr}(F\Sigma) + \frac{1}{4}\mu_g^2\mathrm{tr}^2(F\Sigma) && (12.104)\\
&+\frac{1}{16}(\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma))^2\Big] && (12.105)
\end{aligned}
$$

Hence the variance is

$$
\begin{aligned}
\sigma_z^2 &= \underline{\mu_f^2\mu_g^2}^{(a)} && (12.106)\\
&+\Big[\underline{\mu_f^2\mu_g\mathrm{tr}(G\Sigma) + 2\mu_f\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + \mu_f\mu_g^2\mathrm{tr}(F\Sigma)}^{(b)} && (12.107)\\
&+\mu_f^2\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + 2\mu_f\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + +\mu_g^2\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\Big] && (12.108)\\
&+\Big[\underline{\mu_f\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(G\Sigma)}^{(c)} + \mu_f\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(F\Sigma) && (12.109)\\
&+\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\ \mathrm{tr}(G\Sigma) + \underline{\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(F\Sigma)}^{(d)} && (12.110)\\
&+\underline{\frac{1}{4}\mu_f^2(\mathrm{tr}(G\Sigma))^2}^{(e)} + \mu_f\mathrm{tr}(G\Sigma)\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + \underline{\frac{1}{2}\mu_f\mu_g\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma)}^{(f)} && (12.111)\\
&+\underline{(\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x)^2}^{(g)} + \mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\mathrm{tr}(F\Sigma) + \underline{\frac{1}{4}\mu_g^2\mathrm{tr}^2(F\Sigma)}^{(h)} && (12.112)\\
&+\frac{1}{16}(\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma))^2\Big] && (12.113)\\
&- && (12.114)\\
&\Big[\underline{\mu_f^2\mu_g^2}^{(a)} && (12.115)\\
&+\underline{\big[\mu_f^2\mu_g\mathrm{tr}(G\Sigma) + 2\mu_f\mu_g\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x + \mu_f\mu_g^2\ \mathrm{tr}(F\Sigma)\big]}^{(b)} && (12.116)\\
&+\Big[\underline{\frac{1}{2}\mu_f\mu_g\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma)}^{(f)} + \mu_f\mu_g\mathrm{tr}(F\Sigma G\Sigma) && (12.117)\\
&+\underline{\frac{1}{4}\mu_f^2\mathrm{tr}^2(G\Sigma)}^{(e)} + \underline{\mu_f\ \mathrm{tr}(G\Sigma)\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x}^{(c)} + \underline{\frac{1}{4}\mu_f\mu_g\mathrm{tr}(G\Sigma)\ \mathrm{tr}(F\Sigma)} && (12.118)\\
&+\underline{(\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x)^2}^{(g)} + \underline{\mu_g\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\ \mathrm{tr}(F\Sigma)}^{(d)} + \underline{\frac{1}{4}\mu_g^2\mathrm{tr}^2(F\Sigma)}^{(h)}\Big]\Big] && (12.119)\\
&= \mu_f^2\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + 2\mu_f\mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x + \mu_g^2\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x && (12.120)\\
&+\Big[\mu_f\boldsymbol{g}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\ \mathrm{tr}(F\Sigma) + \mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{f}_x\ \mathrm{tr}(G\Sigma) + \mu_g\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\mathrm{tr}(F\Sigma) + \mu_f\boldsymbol{f}_x^{\mathsf{T}}\Sigma\boldsymbol{g}_x\mathrm{tr}(G\Sigma) && (12.121)\\
&+\frac{1}{16}(\mathrm{tr}(F\Sigma)\mathrm{tr}(G\Sigma))^2 - \mu_f\mu_g\mathrm{tr}(F\Sigma G\Sigma)\Big] && (12.122)
\end{aligned}
$$

hence, we have the two variances of the product $z = \boldsymbol{f}(x)\boldsymbol{g}(x)$

$$\begin{aligned}
\sigma_z^{(2)} &= (\mu_f \boldsymbol{g}_x + \mu_g \boldsymbol{f}_x)^\mathsf{T} \, \Sigma (\mu_f \boldsymbol{g}_x + \mu_g \boldsymbol{f}_x) \,, && (12.123)\\
\sigma_z^{(4)} &= \sigma_z^{(2)} + && (12.124)\\
&\quad + (\mu_f \boldsymbol{g}_x + \mu_g \boldsymbol{f}_x)^\mathsf{T} \, (\mathrm{tr}(F\Sigma)\boldsymbol{g}_x + \mathrm{tr}(G\Sigma)\boldsymbol{f}_x) && (12.125)\\
&\quad + \frac{1}{16}\mathrm{tr}^2(F\Sigma)\mathrm{tr}^2(G\Sigma) - \mu_f\mu_g \mathrm{tr}(F\Sigma G\Sigma) \,. && (12.126)
\end{aligned}$$

13 Bounded Quasi Normal Distribution

Transforming homogeneous vectors to non-homogeneous vectors leads to random variables without mean and variance, if the densities are non-zero on infinite support, e.g. when handling Gaussian random variables. Two remedies are proposed: (1) to limit the support of the given random variables, which corresponds to rejection of outliers, and (2) using first order approximations while avoiding critical configurations.

13.1 Preface

The note (2012) addresses using projective relations, which regularly lead to divisions by the homogeneous part of the homogeneous vector. Assuming a Gaussian distribution for the given entities the result generally, has no finite moments, especially no mean or variance, see Hartley and Zisserman (2000, Sect. A3.1, p.569). We analyse the effect of approximating the Gaussian distribution to a limited range, effectively keeping the moments. The analysis explains, why in most practical cases we do not encounter a dilemma: our input data have no infinite range, as the Gaussian assumption suggests.

13.2 Problem and motivation

Given is a random variable $\underline{x} \sim p_x(x)$ witt $\mu_x = \mathbb{E}(\underline{x})$ and the function $y = 1/x$. The task is to determine the mean value μ_y[1].

Example 13.2.26: ***Side of a rectangle.*** The area A of a rectangular property with the sides a and b is assumed to be given. One side, say a, is observed. Then the other side results from $b = F/a$. Given the mean value of a we want to know the mean value of b. ◇

Example 13.2.27: ***Perspective projektion.***

$$y = \frac{a + bx}{c + dx} \tag{13.1}$$

◇

Example 13.2.28: ***Triangulation from two images.***

$$Z = \frac{Bc}{p_x} \tag{13.2}$$

◇

[1] This is a translation of the note originally written in German.

13.3 Linear and higher approximations

The Taylor series of the function $1/x$ at some point m can be written as

$$\frac{1}{x} = \frac{1}{m} - \frac{x-m}{m^2} + \frac{(x-m)^2}{m^3} + O((x-m)^4) = \sum_{n=0}^{\infty} \frac{-1}{m} \left(\frac{x-m}{-m}\right)^n \tag{13.3}$$

With $m = \mu_x$ we obtain a first approximation

$$\mathbb{E}\left(\frac{1}{x}\right)^{(1)} = \frac{1}{\mu_x} \tag{13.4}$$

a 2nd approximation

$$\mathbb{E}\left(\frac{1}{x}\right)^{(2)} = \frac{1}{\mu_x} - \frac{1}{\mu_x^2} E(x-\mu_x) \stackrel{symm.}{=} \frac{1}{\mu_x} \tag{13.5}$$

and a 3-rd approximation

$$\mathbb{E}\left(\frac{1}{x}\right)^{(3)} = \frac{1}{\mu_x} - \frac{1}{\mu_x^2} E(x-\mu_x) + \frac{1}{\mu_3} E((x-\mu_x)^2) \tag{13.6}$$

$$\stackrel{symm.}{=} \frac{1}{\mu_x}\left(1 + \frac{\sigma_x^2}{\mu_x^2}\right) \tag{13.7}$$

For a symmetric function we only need the even approximations. E.g.

$$\mathbb{E}\left(\frac{1}{x}\right)^{(5)} = \frac{1}{\mu_x} + \frac{1}{\mu_x^3} E((x-\mu_x)^2) + \frac{1}{\mu_x^5} E((x-\mu_x)^4) \tag{13.8}$$

Thus, we only need the even moments of the distribution. If they do not grow faster than μ_x^n, the series may converge.

For a Gaussian the odd moments are zero and the even moments are

$$\mathbb{E}((x-\mu)^2) = 1{\cdot}\sigma_x^2\,, \quad \mathbb{E}((x-\mu)^4) = 1{\cdot}3{\cdot}\sigma_x^4\,, \quad \mathbb{E}((x-\mu)^6) = 1{\cdot}3{\cdot}5{\cdot}\sigma_x^6 \tag{13.9}$$

or for even n

$$\mathbb{E}((x-\mu_x)^n) = (n-1)!!\,\sigma^n\,, \qquad (n-1)!! = (n-1)(n-3){\cdot}...{\cdot}1\,. \tag{13.10}$$

The 5-th approximation thus is

$$\mathbb{E}\left(\frac{1}{x}\right)^{(5)} = \frac{1}{\mu_x}\left(1 + \frac{\sigma_x^2}{\mu_x^2} + \frac{3\,\sigma_x^4}{\mu_x^4}\right) \tag{13.11}$$

Generally, we obtain the odd approximations for a symmetric distribution

$$\mathbb{E}\left(\frac{1}{x}\right)^{(n)} \stackrel{symm.}{=} \frac{1}{\mu_x}\left(1 + \frac{\sigma_x^2}{\mu_x^2} + \frac{3\sigma_x^4}{\mu_x^4} + \frac{15\sigma_x^6}{\mu_x^6} + ...\right) \tag{13.12}$$

$$= \frac{1}{\mu_x} \sum_{i=0:2:n-1} (i-1)!! \left(\frac{\sigma_x}{\mu_x}\right)^i \quad \text{with} \quad n \text{ odd}\,. \tag{13.13}$$

The series is not convergent, since the quotient criterion is not fulfilled, due to the double exponential of n. Therefore, no Taylor expansion can is a good approximation.

13.4 Direct Integration

Another way to show this is direct integration. The mean value of $\underline{y}$ is defined as

$$\mathbb{E}(\underline{y}) = \int_{x=-\infty}^{\infty} \frac{1}{x}\, p_x(x)\, \mathrm{d}x \tag{13.14}$$

Due to the pole at $x = 0$ we separate the integral into the sum of two parts

$$\mathbb{E}(\underline{y}) = \lim_{\epsilon \to 0} \int_{x=-\infty}^{-\epsilon} \frac{1}{x}\, p_x(x)\, \mathrm{d}x + \lim_{\epsilon \to 0} \int_{x=\epsilon}^{\infty} \frac{1}{x}\, p_x(x)\, \mathrm{d}x \overset{p_x(0) \neq 0}{=} -\infty + \infty = \text{unbestimmt} \tag{13.15}$$

If the density $p_x(x)$ at $x = 0$ is not equal to 0, both integrals diverge and the sum is undefined. Hence, for the case $p_x(0) \neq 0$ the mean value of $1/x$ is undefined.

13.5 Discussion

In both derivation we do not obtain a finite mean. A Taylor approximation therefor is no admissible approximation.

The prerequisite for this dilemma is the division by a random variable, whose density is non-zero at the zero of the denominator.

The example with the area 13.2 ony may legitimately argue, that distances are only positive. This argument is not valid for the perspective projection.

13.6 Solution: a bounded distribution for the given observation

If we can bound the distribution of the random variable $\underline{x}$, such that its density is zero at the zero of the denominator, we can solve this dilemma., see Fig. Since, we always

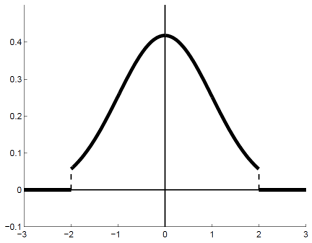


Figure 13.1: Bounded Gaussian, range $[-2, +2]$

perform some type of outlier detection, this may be a reasonable assumption. But we do not want to go to far from the Gaussian distribution

Therefore, we want to have a density $p(x \mid k)$ which fulfil the following conditions

$$\int_{-k}^{+k} p(x)\mathrm{d}x = 1 \tag{13.16}$$

$$\int_{-k}^{+k} xp(x)\mathrm{d}x = 0 \tag{13.17}$$

$$\int_{-k}^{+k} x^2 p(x)\mathrm{d}x = 1 \tag{13.18}$$

$$\int_{-k}^{+k} x^3 p(x)\mathrm{d}x = 0 \tag{13.19}$$

$$\int_{-k}^{+k} x^4 p(x)\mathrm{d}x = 3 \tag{13.20}$$

This is a density restricted to the range $[-k, +k]$ whose first four moments are identical to those of the Gaussian distribution, sometimes called quasi-Gaussian.

We thus obtain densities depending on the chosen range parameter k.

13.7 Maximum Entropy Distribution

We start from the n constraints

$$\int_{-k}^{+k} f_i(x)p(x)\mathrm{d}x \quad = \quad 1 \qquad f_i(x) = x^i \qquad i = 0, ..., n \tag{13.21}$$

If we follow the principle of maximum entropy, we obtain the density in the form (Koch, 1990, (223.14))

$$p(x) = \exp\left(-\sum_{i=0}^{n} k_i f_i(x)\right) \quad \text{with} \quad f_0(x) = 1\,, \tag{13.22}$$

from which we may determine the parameters k_i.

If we restrict to moments up to 2nd order we obtain the density

$$p(x \mid \mu, \sigma^2) = \exp\left(-k_0 - k_1 x - k_2 x^2\right) \qquad x \in [-k, +k] \tag{13.23}$$

which is a bounded normal distribution. Since the mean value should be 0, we have $k_1 = 0$. Therefore we obtain

$$p(x) = \frac{\exp(-\frac{1}{2}x^2/s^2)}{Z} \tag{13.24}$$

by choosing s and Z such that the conditions (13.16) and (13.18) are fulfilled.

From the first constraint

$$\int_{-k}^{+k} p(x)\mathrm{d}x = \int_{-k}^{+k} \frac{\exp\left(-\frac{1}{2}\frac{x^2}{s^2}\right)}{Z}\mathrm{d}x = 1 \tag{13.25}$$

we obtain

$$Z(s, k) = \mathrm{erf}\left(\frac{k}{\sqrt{2}s}\right) \tag{13.26}$$

For $k \gg s^2$ this normalization constant is close to 1.

The second constraint for s now is

$$s^2\left(1 - \frac{2k}{Z}\underbrace{\frac{1}{\sqrt{2\pi s^2}}\exp\left(-\frac{1}{2}\frac{k^2}{s^2}\right)}_{<<1}\right) = \sigma^2\,. \tag{13.27}$$

For a given variance $\sigma^2 = 1$ and the range parameter k we can derive s numerically. Für $k \gg s^2$ we find $s \approx (1 + \varepsilon)\sigma$ with a generally small value ε, see Tab. 13.1. The density

k	Z	s	$p(x = k)$
3	.9968686297	1.015386985	.005012998163
4	.9999360767	1.000540144	.0001349266451
5	.9999994267	1.000007435	.000001486985673

Table 13.1: Normierungsfaktoren Z, Steuungen s und Randwerte $p(k)$ für $\sigma = 1$.

$p(x \mid \sigma^2, k)$ at $x = k$ is not continuous. The jump decreases with increasing k.

The even moments of the normal density and of $p(x \mid \sigma = 1, k = 4)$ are listed in the Tab. 13.2 For small n the moments only differ slightly. For large n the moments of the approximation are much smaller than those of the normal density, why the series for the mean value converge.

n	Gauss	$p(x \mid 1, 4)$	q
2	1	1.000000000	1.000000
4	3	2.985952469	0.995317
6	15	14.66926678	0.977951
...			
20	654,729,075	151,821,743.5	0.231822
40	319,830,986,772,877,770,815,625	49,961,967,754,647,038,660.8	0.000156

Table 13.2: Moments m_n and m_n^a of the normal density and of its approximation $p(x \mid , \sigma = 1, k = 4)$ together with the ratio $q(n) = m_n^a / m_n$.

13.8 Relevance

The common stochastical model for observations is the Gaussian distribution. With equal justification we could choose a bounded normal distribution. Both distributions differ only marginally.

When performing sums – possibly weighted – of many observations the range of the resulting distribution increases with the number of observations, however, restricting its range to a value depending on its standard deviation would lead to acceptable approximations.

When dividing by random variables, we need to make sure, that the range of the distribution does not contain a zero of the denominator. Otherwise we obtain results without mean and variance. A classic example is the division of variables with standard normal distribution: the ratio is Cauchy distributed, with no mean and variance.

This analysis explains, why in most practical cases we do not encounter the dilemma: our input data have no infinite range, as the Gaussian assumption suggests.

14 Precision of the Inverse of an Uncertain Matrix

We derive the covariance matrix of the inverse of a matrix of random variables, whose covariance matrix is given, assuming the given covariance matrix is regular. The covariance matrix of the given matrix may be singular, allowing to handle transformation matrices with a low degree of freedom.

14.1 The Goal

Let the stochastic variables $\underline{a}_{ij}$ be collected in the matrix $\underline{A}$ or, equivalently, in the vector $\underline{\boldsymbol{a}} = \mathrm{vec}(\underline{A})$ with covariance matrix $\mathbb{D}(\underline{\boldsymbol{a}}) = \Sigma_{aa}$. Following the proof, the covariance matrix Σ_{bb} of the elements $\underline{b}_{ij}$ of its inverse $\underline{B} = \underline{A}^{-1}$

$$\mathbb{D}(\underline{\boldsymbol{b}}) = \Sigma_{bb} = (\tilde{A}^{-\mathsf{T}} \otimes \tilde{A}^{-1})\Sigma_{aa}(\tilde{A}^{-1} \otimes \tilde{A}^{-\mathsf{T}})$$

where the tilde here indicates the mean value $\tilde{A} := \mathbb{E}(\underline{A})$.

14.2 Proof

We have

$$A\,B = I$$

thus

$$\mathrm{d}A\,B + A\,\mathrm{d}B = \mathbf{0}$$

or

$$\mathrm{d}B = -B\,\mathrm{d}A\,B$$

With

$$\mathrm{d}\boldsymbol{a} = \mathrm{vec}(\mathrm{d}A) \qquad \text{and} \qquad \mathrm{d}\boldsymbol{b} = \mathrm{vec}(\mathrm{d}B)$$

we obtain

$$\mathrm{d}\boldsymbol{b} = -(B^{\mathsf{T}} \otimes B)\,\mathrm{d}\boldsymbol{a}$$

yielding the result.

Remark: In case we use the *rows* of the matrices, thus

$$\mathrm{d}\boldsymbol{a}' = \mathrm{vec}(\mathrm{d}A^{\mathsf{T}}) \qquad \text{and} \qquad \mathrm{d}\boldsymbol{b}' = \mathrm{vec}(\mathrm{d}B^{\mathsf{T}})$$

we obtain

$$\mathrm{d}\boldsymbol{b}' = -(B \otimes B^{\mathsf{T}})\mathrm{d}\boldsymbol{a}'$$

and therefore

$$\Sigma_{bb} = (\tilde{B} \otimes \tilde{B}^{\mathsf{T}})\Sigma_{aa}(\tilde{B}^{\mathsf{T}} \otimes \tilde{B})$$

15 Remarks on the Equivalence of Gauge- or S-transformations and Reducing Homogeneous Coordinates

Gauge-transformations change the reference coordinate of the covariance matrix of the coordinates of a point cloud, leave the relative uncertainty of the points invariant, and result in a regular covariance matrix. Similarly, reducing uncertain homogeneous entities with a possibly regular covariance matrix to a tangent space, specified by some constraint, leads to what can be called reduced coordinates. These reduced coordinates have a regular covariance matrix and capture the full information of the uncertainty of the original homogeneous coordinates. We show that both concepts are equivalent.

15.1 Preface

This note (2016) adresses the relation between S-transformations (Baarda, 1973) for fixing the gauge during estimation and using reduced coordinates (Förstner, 2010) for estimating homogeneous entities without constraints. They are shown to be based on the same concept, namely controlling or eliminating the the null space of entities with singular covariance matrices, where the – possible non-linear – constraints are known.

15.2 Motivation and Goal

We assume we have a random U-vector $\underline{\boldsymbol{x}}$ with covariance matrix Σ_{xx}. The random variables $\underline{x}_u$ are constrained by the H algebraically independent nonlinear constraints $\boldsymbol{h}(\boldsymbol{\mu}_x) = \mathbf{0}$. Examples are

- coordinates estimated in a free geodetic network or a free bundle adjustment, thus, without using reference points to fix the coordinate system; here the constraints fix the gauge of the covariance matrix (cf. Förstner and Wrobel, 2016, Sect. 4.5.3).
- homogeneous coordinates or matrices to be used in geometric reasoning (cf. Förstner and Wrobel, 2016, Sect. 10.2.2.1, (10.26)); here the constraints fix the scale or enforce other internal properties, such as the singularity of the fundamental matrix (cf. Förstner and Wrobel, 2016, Sect. 13.3.4.1).

Figure 15.1: Left: Regular S-transformation of 4 2D points with full covariance matrix fixing the coordinates of the first point (translation only). **Right:** Uncertain homogeneous vector in $\mathbb{P}^2$ reduced to the uncertain 1D point on S^2

s. Fig. 15.1

S-transformations according to Baarda (1973) (1) can be used to enforce the covariance matrix to have a prespecified gauge and (2) are required to evaluate the covariance matrix w.r.t. a criterion matrix.[1]

Reduced coordinates according to Förstner (2010) can be used (1) for taking $\boldsymbol{x}$ as observations in an estimation procedure, (2) for deriving estimates $\hat{\boldsymbol{x}}$ from observations, and (3) for testing hypotheses about $\boldsymbol{\mu}_x$.

The goal of this note is to show that the regular S-transformation and reducing the coordinates are equivalent w.r.t. evaluating the covariance matrix of a constrained random vector.

15.3 S-Transformations

We distinguish singular and regular S-transformations. Singular S-transformations yield singular covariance matrices with a specified gauge. Regular S-transformations yield regular covariance matrices for a subset of the parameters.

As an example, for a set of points in $\mathbb{R}^3$ which are observed by directions or angles only we have the seven constraints which fix the gauge w.r.t. given approximate values $\boldsymbol{x}^a$:

- The position of the coordinate system; these are three gauge parameters; this could be enforced by the constraints

$$\boldsymbol{d} = 0 \quad \text{with} \quad \boldsymbol{d} = \sum_i w_i(\boldsymbol{x}_i - \boldsymbol{x}_i^a) \tag{15.1}$$

  where $\boldsymbol{x}_i^a$ are fixed approximate values and w_i are weights $\in [0, 1]$.

- The rotation of the coordinate system; these are three gauge parameters; this could be enforced by the constraints

$$R = I \quad \text{with} \quad R = \texttt{mean\_rotation}(\{\boldsymbol{x}_i, \boldsymbol{x}_i^a, w_i\}) \tag{15.2}$$

  where the weighted mean of the rotation is derived from Algorithm 12 in (cf. Förstner and Wrobel, 2016, Sect. 10.5.4.2).

- The scale of the coordinate system; this is one gauge parameter; it could be enforced from the constraint

$$s = 1 \quad \text{with} \quad \log s = \sum_i w_i \log \frac{|\boldsymbol{x}_i - \boldsymbol{x}_0|}{|\boldsymbol{x}_i^a - \boldsymbol{x}_0^a|} \tag{15.3}$$

  where $\boldsymbol{x}_0$ and $\boldsymbol{x}_0^a$ are the weighted centroids of the points.

[1] We only discuss transformations of the covariance matrices, hence, differential coordinate transformations, which leave the mean values of the random variables unchanged. When comparing the result of two different estimates for the same parameters (point coordinates), we assume, they already have been transformed by a K-transformation according to the notion by Molenaar (1981).

We need the Jacobian of the constraints

$$\underset{U\times H}{H} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}^\mathsf{T}}\,, \tag{15.4}$$

and possibly a weight matrix, which defines which parameters take part in the fixation of the gauge of the covariance matrix

$$W = \mathrm{Diag}([w_u])\,, \quad \text{with} \quad w_u \in [0,1]\,, \tag{15.5}$$

here taken for each parameter x_u individually.

15.3.1 Singular S-Transformation

The *singular S-transformation* uses the $U \times U$ S-matrix (cf. Förstner and Wrobel, 2016, (4.223))

$${}^cS = I_U - H(H^\mathsf{T}{}^cW H)^{-1}H^\mathsf{T}{}^cW\,, \tag{15.6}$$

which is a projection matrix. It is indexed by c, indicating which weight matrix ${}^cW$ is used, It has null space H since

$${}^cSH = 0\,. \tag{15.7}$$

The transformed coordinates, also indexed with c, then are

$${}^c\underline{\Delta \boldsymbol{x}} = {}^cS\ \underline{\Delta \boldsymbol{x}} \tag{15.8}$$

with covariance matrix

$${}^c\Sigma_{xx} = {}^cS\ \Sigma_{xx}\ {}^cS^\mathsf{T}\,. \tag{15.9}$$

Observe, we would obtain this covariance matrix also when enforcing the constraints

$$H^\mathsf{T}{}^cW\underline{\boldsymbol{x}} = \mathbf{0}\,. \tag{15.10}$$

Therefore

$$H^\mathsf{T}{}^cW{}^c\Sigma_{xx} = 0\,. \tag{15.11}$$

15.3.2 Regular S-Transformation

The regular S-transformation uses a special choice of W, namely

$${}^kW = \mathrm{Diag}([w_u]) \quad \text{with} \quad \sum_u w_u = H\,. \tag{15.12}$$

Without loss of generality we can assume that it has the form

$${}^kW = \mathrm{Diag}([\underbrace{1,...,1}_{H\ \text{ones}}, \underbrace{0,...,0}_{U-H\ \text{zeros}}])\,. \tag{15.13}$$

The gauge specified by the constraints onto the first H parameters. Thus the upper $H \times H$-submatrix H_1 of H is assumed to be regular. We therefore partition $\boldsymbol{x}$, H and W

$$\boldsymbol{x} = \begin{bmatrix} \underset{U\times 1}{\boldsymbol{x}_1} \\ \underset{U\times 1}{\boldsymbol{x}_2} \end{bmatrix}\,, \qquad \underset{U\times H}{H} = \begin{bmatrix} \underset{H\times H}{H_1} \\ \underset{(U-H)\times H}{H_2} \end{bmatrix} \quad \text{and} \quad \underset{U\times U}{W} = \begin{bmatrix} I_H & 0 \\ 0 & \underset{(U-H)\times(U-H)}{0} \end{bmatrix} \tag{15.14}$$

The singular S-transformation matrix S then has the following structure (for K, cf. the proof below)

$${}^kS = \begin{bmatrix} 0_{H\times H} & 0 \\ K & I_{U-H} \end{bmatrix} \quad \text{with} \quad K = -H_2H_1^{-1}\,. \tag{15.15}$$

Then the transformed vector has the structure

$$ {}^k\underline{\Delta \boldsymbol{x}} = {}^k S\, \underline{\Delta \boldsymbol{x}} = \begin{bmatrix} \mathbf{0}_H \\ {}^k\underline{\Delta \boldsymbol{x}}_r \end{bmatrix} \tag{15.16} $$

Hence, the first H elements of ${}^k\underline{\Delta \boldsymbol{x}}$ are fixed values; the remaining parameters, called ${}^k\underline{\Delta \boldsymbol{x}}_r$, capture the complete uncertainty of ${}^c\underline{\Delta \boldsymbol{x}}$. Thus the covariance matrix has the structure

$$ {}^c\Sigma_{xx} = \begin{bmatrix} 0 & 0 \\ 0 & {}^c\Sigma_{xx,r} \end{bmatrix} . \tag{15.17} $$

We arrive at the *regular S transformation* by omitting the fixed elements of ${}^c\underline{\Delta \boldsymbol{x}}$. In order to arrive at an explicit expression for this covariance matrix we define the regular S-matrix, now indexed with k,

$$ \underset{(U-H)\times U}{{}^k S_r} = [0_{(U-H)\times H}\,,\, I_{U-H}]\; {}^k S = [-H_2 H_1^{-1}, I_{U-H}] \tag{15.18} $$

to obtain the regularly transformed coordinates

$$ {}^k\underline{\Delta \boldsymbol{x}}_r = {}^k S_r\; \underline{\Delta \boldsymbol{x}} = {}^k\underline{\Delta \boldsymbol{x}}_r = \underline{\Delta \boldsymbol{x}}_2 - H_2 H_1^{-1} \underline{\Delta \boldsymbol{x}}_1 \,, \tag{15.19} $$

with its regular covariance matrix

$$ {}^k\Sigma_{xx,r} = {}^k S_r\; \Sigma_{xx}\; {}^k S_r^{\mathsf{T}} \,. \tag{15.20} $$

Proof: We derive K. Omitting the index c we have

$$ S = \begin{bmatrix} I_H & 0 \\ 0 & I_{U-H} \end{bmatrix} - \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} (H_1^{\mathsf{T}} H_1)^{-1} [H_1^{\mathsf{T}}, 0] \tag{15.21} $$

$$ = \begin{bmatrix} 0 & 0 \\ -H_2 H_1^{-1} & I_{U-H} \end{bmatrix} . \tag{15.22} $$

Hence, $K = -H_2 H_1^{-1}$.

15.4 Reduced Coordinates

Reduced coordinates $\boldsymbol{x}_r$ result from projecting the random vector $\underline{\boldsymbol{x}}$ to the tangent space of the manifold $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ at $\boldsymbol{\mu}_x$ (cf. Förstner and Wrobel, 2016, Figs. 10.4 and 10.10, Sect. 10.2.2.1 and Sect. 10.6.1). The tangent space is spanned by the null space of H^{T}. This null space is not unique. We assume it is spanned by a set of H orthonormal vectors collected in the matrix

$$ \underset{U\times(U-H)}{J_r(\boldsymbol{\mu}_x)} = \mathsf{null}(H^{\mathsf{T}}(\boldsymbol{x}))\Big|_{x=\mu_x} \quad \text{with} \quad J_r^{\mathsf{T}}(\boldsymbol{\mu}_x) J_r(\boldsymbol{\mu}_x) = I_H \,. \tag{15.23} $$

The reduced vector $\underline{\boldsymbol{x}}_r$ is then defined as the projection of $\underline{\boldsymbol{x}}$ onto the tangent space,

$$ \underline{\boldsymbol{x}}_r = J_r^{\mathsf{T}}(\boldsymbol{\mu}_x)(\underline{\boldsymbol{x}} - \boldsymbol{\mu}_x) \,, \tag{15.24} $$

and has covariance matrix

$$ \Sigma_{xx,r} = J_r^{\mathsf{T}}(\boldsymbol{\mu}_x)\, \Sigma_{xx}\, J_r(\boldsymbol{\mu}_x) \,. \tag{15.25} $$

15.5 Comparison

Both vectors, the vector ${}^k\underline{\Delta \boldsymbol{x}}_r$ from the regular S-transformation and the vector $\underline{\boldsymbol{x}}_r$ from the reduction, lie in the null space of $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$. This is because $J_r(\boldsymbol{\mu}_x)$ and ${}^k S_r$ span the null space of H^{T}:

$$ H^{\mathsf{T}} J_r(\boldsymbol{\mu}_x) = H^{\mathsf{T}}\, {}^k S_r = 0 \,. \tag{15.26} $$

The second relation results from

$$[H_1^\mathsf{T} H_2^\mathsf{T}] \begin{bmatrix} -H_1^{-\mathsf{T}} H_2^\mathsf{T} \\ I_{U-H} \end{bmatrix} = -H_2^\mathsf{T} + H_2^\mathsf{T} = 0\,. \tag{15.27}$$

Therefore, both covariance matrices of the transformed vector and the reduced vector capture the full uncertainty of $\underline{x}$.

Thus given two covariance matrices which are to be compared, the comparison leads to the same result with either transformation. Hence, given two covariance matrices C and Q and for brevity using $S := {}^k S_r$, the generalized eigenvalues,

$$\boxed{\lambda(S^\mathsf{T} C S, S^\mathsf{T} Q S) = \lambda(J_r^\mathsf{T} C J_r, J_r^\mathsf{T} Q J_r)\,,} \tag{15.28}$$

are identical.

Proof: The two matrices S and J_r have full rank and span the nullspace of H^T. Thus there is a regular transformation A such that

$$J_r = SA\,. \tag{15.29}$$

Therefore comparing we need to show, that the generalized eigenvalues of

$$\lambda(S^\mathsf{T} C S, S^\mathsf{T} Q S) \quad \text{and} \quad \lambda(A^\mathsf{T} S^\mathsf{T} C S A, A^\mathsf{T} S^\mathsf{T} Q S A) \tag{15.30}$$

are identical. This is because the second term can be replaced by

$$\lambda(A^\mathsf{T} S^\mathsf{T} C S A (A^\mathsf{T} S^\mathsf{T} Q S A)^{-1}) = \lambda(S^\mathsf{T} C S A A^{-1} (S^\mathsf{T} Q S)^{-1} A^{-\mathsf{T}} A^\mathsf{T}) = \lambda(S^\mathsf{T} C S, S^\mathsf{T} Q S) \tag{15.31}$$

which is identical to the first term. ◇

15.6 Conclusions

The concept of S-transformation has been developed for analysing sets of points in 2D or 3D, thus for a specific class of random vectors. It is very flexible, as the gauge can be controlled using the weight matrix W or – equivalently – by selecting those parameters which define the gauge. The advantage of the S-transformation is, that an explicit expression for the tangent space (namely the null space of H^T) is available for the main applications.

The concept of reduced parameters has been developed for simplifying estimation with random vectors, which are constrained. It is a general technique, as it works for all types of constraints. The disadvantage of reduced coordinates is that we have an explicit expression for the tangent space (i.e., the matrix J_r) only for specific constraints, but for large random vectors the determination of the tangent space is computationally prohibitive. However, for sets of vectors with only a few elements, say up to 12, the method leads to efficient algorithms, e.g., in bundle adjustment.

16 Gauge Choice and Loop Closing of Uncertain Polygons

We present a Cincerella animation for exploring the effect of choosing a specific gauge/datum and of loop closing onto an uncertain polygon. The note provides the necessary derivations consistently using complex numbers for representing 2D points and observed distance ratios and angles.

16.1 Scope

We provide the basics for deriving the uncertainty of an open or closed 2D polygon by using a Gauss–Markov model with constraints for fixing the gauge/datum. We assume isotropic uncertainty and therefore can use complex numbers for representing point and measurements to advantage.

We follow the chapter *A Generalization of the concept strength of figure* in (Baarda, 1968) and use complex numbers for 2D points and observed logarithms of lenght ratios and of angles. We only provide the relations which are necessary to define the Gauss-Markov model for an open or closed polygon as a free network.

16.2 The basic relations

Points (x, y) in the plane may be represented by complex numbers $z = x + \mathrm{i}y$. We assume their uncertainty is is circular symmetric, i.e., isotropic. Furthermore, we assume the point cloud is defined up to a similar transformation. Then angles and distance ratios are invariant quantities.

We describe the complex entities involved, present the functional and stochastical relations between the observations and the points, the Gauss–markov model for the free polygon, and the uncertainty of a point w.r.t. to a pair of points.

16.2.1 Complex entities

The chaining in a polygon $(\boldsymbol{z}_1, ..., \boldsymbol{z}_I)$ based on onserved distance ratios and angles, i.e., bearing/direction differences of neighbouring points

$$r_{jik} = \frac{d_{ik}}{d_{ij}} \quad \text{and} \quad \alpha_{jik} = \beta_{ik} - \beta_{ij} \tag{16.1}$$

can be expressed as

$$\boldsymbol{z}_k = \boldsymbol{z}_i + r_{jik} R(\alpha_{jik}) \, (\boldsymbol{z}_j - \boldsymbol{z}_i) \quad \text{with} \quad (j,i,k) = (i-1, i, i+1) \,. \tag{16.2}$$

see Fig. 16.1.

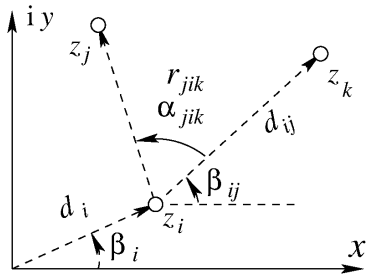


Figure 16.1: Entities for one, two and three points. Coordinates $\boldsymbol{z}_i = d_i R(\beta_i)\boldsymbol{e}_1$, coordinate differences $\boldsymbol{z}_{ij} = \boldsymbol{z}_j - \boldsymbol{z}_i = d_{ij} R(\beta_{ij})\boldsymbol{e}_1$ between j and i, local scaled rotation $\boldsymbol{z}_{ik} = r_{jik} R(\alpha_{jik})\, \boldsymbol{z}_{ij}$ between segment (ji) and segment (ik). The vector $\boldsymbol{e}_1 = [1,0]^\mathsf{T}$ is the first unit vector

We will use complex entities to express this relation, by integrating the distance ratios and the angels into a complex number, similar to the point coordinates.

We use the entities in Tab. 16.1. In all cases we assume indices indicating the order of the entity. Obviously, when analysing the relation between three point, the logarithm

	Name	real representation	complex representation
	point		
1	coordinates	$\boldsymbol{z} = [x, y]^\mathsf{T}$	$z = x + \mathrm{i}y$
2	distance d to origin	$d = \lvert\boldsymbol{z}\rvert$	$d = \lvert z\rvert$
3	bearing/direction β	$\beta = \operatorname{atan2}(y, x)$	$\beta = \arg(z)$
4	polar coordinates	$\boldsymbol{z} = d\, R(\beta)\, \boldsymbol{e}_1$	$z = d\, \mathrm{e}^{\mathrm{i}\beta} = \mathrm{e}^{\ln d + \mathrm{i}\beta}$
	point pair		
5	local translation	$\boldsymbol{z}_j = \boldsymbol{z}_i + \boldsymbol{z}_{ij}$	$z_j = z_i + z_{ij}$
6	coordinate difference	$\boldsymbol{z}_{ij} = \boldsymbol{z}_j - \boldsymbol{z}_i$	$z_{ij} = z_j - z_j = \mathrm{e}^{\ln d_{ij} + \mathrm{i}\beta_{ij}}$
7	distance d	$d_{ij} = \lvert\boldsymbol{z}_{ij}\rvert$	$d_{ij} = \lvert z_{ij}\rvert$
8	bearing/direction β	$\beta_{ij} = \operatorname{atan2}(y_{ij}, x_{ij})$	$\beta_{ij} = \arg(z_{ij})$
9	local polar coordinates	$\boldsymbol{z}_{ij} = d_{ij} R(\beta_{ij})\, \boldsymbol{e}_1$	$\Lambda_{ij} = \ln z_{ij} = \ln d_{ij} + \mathrm{i}\beta_{ij}$
	point triplet		
10	local scaled rotation	$\boldsymbol{z}_{ik} = r_{jik} R(\alpha_{jik})\, \boldsymbol{z}_{ij}$	$z_{ik} = \mathrm{e}^{\Pi_{jik}} = \mathrm{e}^{\ln r_{jik} + \mathrm{i}\alpha_{jik}}\, z_{ij}$ $\Pi_{jik} = \Lambda_{ik} - \Lambda_{ij} = \ln \frac{z_{ik}}{z_{ij}}$
11	angle	$\alpha_{jik} = \delta_{ik} - \delta_{ij}$	$\alpha_{jik} = \beta_{ij} - \beta_{ik} = \Im(\Pi_{jik})$
12	scale ratio	$r_{jik} = \frac{d_{ik}}{d_{ij}}$	$\ln r_{jik} = \ln d_{ik} - \ln d_{ij} = \Re(\Pi_{jik})$
	similarity		
13	transformation	$\boldsymbol{z}' = mR(\phi)\boldsymbol{z} + \boldsymbol{t}$	$z' = sz + t$
14	scaled rotation	$mR(\phi)$	$s = \mathrm{e}^{\log m + \mathrm{i}\phi}$

Table 16.1: Names, real and complex entities for points, point pairs, point triplets and similarities

of scale ratios play the same role as angles. If in a polygon the local scaled rotation

$$\Pi_{jik} = \ln r_{jik} + \mathrm{i}\alpha_{jik} \tag{16.3}$$

is observed for all $i > 1$, the chaining of the points follows from

$$z_k = z_i + \mathrm{e}^{\Pi_{jik}} \, (z_j - z_i) \quad \text{with} \quad (j,i,k) = (i-1, i, i+1) \,. \tag{16.4}$$

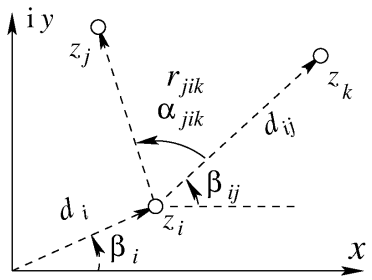


Figure 16.2: Entities for one, two and three points. Coordinates $z_i = d_i \exp \mathrm{i}\beta_i$, coordinate differences $z_{ij} = z_j - z_i = d_{ij} \exp(\mathrm{i}\,\beta_{ij})$ between i and j, local scaled rotation $z_{ik}/z_{ij} = r_{jik} \exp(\mathrm{i}\,\alpha_{jik})$ between segment (ji) and segment (ik)

16.2.2 Observing a polygon

Therefore, we assume a polygon $(z_1, z_2, ..., z_I)$ is observed by the $I-2$ local scaled rotations $\Pi_{jik}(r_{jik}, \alpha_{jik})$ at all points, except for the first and the last one. In case the polygon is closed, we in addition have two additional scaled ratations $\Pi_{I,1,2}$ and $\Pi_{I-1,I,1}$ for the first and the last point.

We assume, the relative accuracy σ_r/r of the scale ratios and standard deviations σ_α of the angles are identical. This is equivalent to assuming the logarithm $\ln r_{jik}$ of the scale ratios and the angles α_{jik} have the same standard deviation, say σ_i, since from

$$q = \ln r \quad \text{and} \quad \frac{\mathrm{d}q}{\mathrm{d}r} = \frac{1}{r} \tag{16.5}$$

we obtain

$$\sigma_q = \sigma_{\ln r} = \frac{\sigma_r}{r}\,. \tag{16.6}$$

Hence, we assume

$$\mathbb{D}\left(\begin{bmatrix} \ln \underline{r}_{jik} \\ \underline{\alpha}_{jik} \end{bmatrix}\right) = \sigma_i^2 I_2\,. \tag{16.7}$$

Hence, we assume isotropy. When using the combined entity Π_{jik} its distribution is circular symmetric with variance $2\sigma^2$, see https://en.wikipedia.org/wiki/Complex_random_variable. Hence we have

$$\mathbb{V}(\Re(\underline{\Pi}_{jik})) = \mathbb{V}(\Im(\underline{\Pi}_{jik})) = \frac{1}{2}\mathbb{V}(\underline{\Pi}_{jik}) \quad \text{and} \quad \mathbb{C}\mathrm{ov}(\Re(\underline{\Pi}_{jik}), \Im(\underline{\Pi}_{jik})) = 0\,. \tag{16.8}$$

In the following we call the combined value

$$\Pi_{jik} = \ln r_{jik} + \mathrm{i}\alpha_{jik} \tag{16.9}$$

the scaled rotation, taking into account, that the name omits the logarithm of the scale ratio of neighbouring sides.

Remark: This model fits to the idea that the path along the polygon is observed with a camera, and the relative pose between two points (ij) and the next point k is determined by evaluating image triplet, which provide the scale ratios r_{jik} and the angles α_{jik} with approximately the same (relative) accuracy. ◇

16.2.3 The Gauss–Markov model for the free polygon

For establishing a Gauss–Markov model with gauge/datum constraints we need the linearization of the observed entities.

16.2.3.1 Observation equations

We assume the local observations are scaled rotations of subsequent points in the polygon:

$$\Pi_{jik} \quad \text{with} \quad (j,i,k) = (i-1, i, i+1) \quad \text{and} \quad i = 2, ..., I-1\,. \tag{16.10}$$

In case of a closed polygon, representing a loop closing, we have the additional observations

$$\Pi_{jik} \quad \text{with} \quad (j,i,k) = (I,1,2) \quad \text{and} \quad \Pi_{jik} \quad \text{with} \quad (j,i,k) = (I-1, I, 1) \quad \text{for} \quad i = 1, I\,. \tag{16.11}$$

for the starting and the end point of the polygon. The complete set of observations $\boldsymbol{y}$ then consists of the complex values

$$\boldsymbol{y} = [y_i] := [\Pi_{jik}]\,, \tag{16.12}$$

with or without the border observations.

The observation equations are

$$\boxed{\mathbb{E}(\underline{\Pi}_{jik}) = \ln \frac{z_k - z_i}{z_j - z_i}\,.} \tag{16.13}$$

which gives an explicit relation to the unknown point coordinates $z_i, i = 1, ..., I$. The stochastical model is

$$\boxed{\mathbb{D}\left(\underline{\Pi}_{jik}\right) = 2\sigma_i^2\,.} \tag{16.14}$$

where σ_i represents the standard deviations of the real and the imaginary part of Π_{jik}, i.e., the logarithm of the distance ratios and the angles.

16.2.3.2 Gauge constraints

Since we we can determine the coordinates only up to a similarity, we need to enforce constraints onto the coordinates. This fixes the gauge or the datum of the uncertainty of the points. The constraints are related to the approximate values z_i^0 of the points. We require, that the mean of all coordinates and the mean scaled rotation around the centroid $\overline{z}$ are zero. Since the scaled rotation is represented as a complex variable $\log s + \mathrm{i}\alpha$, requiring that the scaled rotation is 0, is equivalent to the requirement, that the average scale ratio to the approximate values z_i^0 is 1. Now we observe, that the centerd coordinates are

$${}^c z_i = z_i - \overline{z} = \frac{\sum_i z_i^0}{I} \tag{16.15}$$

This leads to the constraints for the estimated coordinates, now referring to the centroid

$$\boxed{c_1 = \textstyle\sum_i w_i(\widehat{z}_i - \overline{z}) = 0 \quad \text{and} \quad c_2 = \sum_i w_i(s\,(\widehat{z}_i - \overline{z}) + t - z_i^0) = 0\,,} \tag{16.16}$$

the first constraint requiring the weighted mean of at least two point estimates being 0, the second constraint requiring, that the weighted mean difference between the (with s and t) transformed estimated and the approximate values is 0.

Linearization of the observation equations and the constraints lead to a linearized Gauss–Markov model.

16.2.4 Linearization

16.2.4.1 Linearized observation equations

Linearization of the observation equations leads to

$$\Pi_{ijk} + v_{jik} = \Pi_{ijk}^0 + \boldsymbol{a}_i^\mathsf{T}\boldsymbol{\theta} \quad \text{with} \quad \mathbb{E}(\underline{\Pi}_{jik}) = \ln \frac{z_k - z_i}{z_j - z_i} \tag{16.17}$$

with the non-zero elements of $\boldsymbol{a}_i$

$$\boldsymbol{a}_{i0}^{\mathsf{T}} = \left[\frac{\partial \Pi_{ijk}}{\partial z_j} \ , \frac{\partial \Pi_{ijk}}{\partial z_i} , \frac{\partial \Pi_{ijk}}{\partial z_k} \right] \tag{16.18}$$

and the unknown parameters

$$\boldsymbol{\theta} := \begin{bmatrix} z_1 \\ \dots \\ z_i \\ \dots \\ z_I \end{bmatrix} . \tag{16.19}$$

The index 0 in $\boldsymbol{a}_{i0}^{\mathsf{T}}$ indicates that only the non-zero elements are shown.

The differentials of the various elements are given as follows

$$\begin{aligned}
\mathrm{d}z_{ij} &= \mathrm{d}z_j - \mathrm{d}z_i && (16.20)\\
\mathrm{d}\Lambda_{ij} &= \mathrm{d}\ln z_{ij} && (16.21)\\
&= \frac{1}{z_{ij}}\mathrm{d}z_{ij} = \frac{1}{z_{ij}}(\mathrm{d}z_j - \mathrm{d}z_i) && (16.22)\\
\mathrm{d}\Pi_{jik} &= \mathrm{d}(\Lambda_{ik} - \Lambda_{ij}) && (16.23)\\
&= \frac{1}{z_{ik}}(\mathrm{d}z_k - \mathrm{d}z_i) - \frac{1}{z_{ij}}(\mathrm{d}z_j - \mathrm{d}z_i) && (16.24)\\
&= \underbrace{\left[-\frac{1}{z_{ij}} , \frac{1}{z_{ij}} - \frac{1}{z_{ik}} , \frac{1}{z_{ik}} \right]}_{da_{i0}^{\mathsf{T}} = [a_{ij}, a_{ii}, a_{ik}]} \begin{bmatrix} \mathrm{d}z_j \\ \mathrm{d}z_i \\ \mathrm{d}z_k \end{bmatrix} && (16.25)\\
&= \left[\frac{1}{z_{ji}} , -\frac{1}{z_{ji}} - \frac{1}{z_{ik}} , \frac{1}{z_{ik}} \right] \begin{bmatrix} \mathrm{d}z_j \\ \mathrm{d}z_i \\ \mathrm{d}z_k \end{bmatrix} && (16.26)
\end{aligned}$$

the last expression mimiking the second derivative operator $[1, -2, 1]$, which would result from a straight polygon, say in x-direction with all distances being one.

The complete Jacobian of an open polygon is a tridiagonal matrix

$$A_{\text{inner}} = \mathrm{Tri}([a_{i-1,i}, a_{i,i}, a_{i,i+1}]) , \quad \text{for} \quad i = 2, ..., I-1 . \tag{16.27}$$

and for a closed polygon may be augmented by

$$A_{\text{border}} = \begin{bmatrix} a_{I-1,1} & a_{I-1,2} & 0 & 0 & 0 & 0 & a_{I-1,I} \\ a_{I,1} & 0 & 0 & 0 & 0 & a_{I,I-1} & a_{I,I} . \end{bmatrix} \tag{16.28}$$

to obtain the Jacobian $A = \partial \boldsymbol{y} / \partial \boldsymbol{\theta}$.

16.2.4.2 Linearized constraints

The linearization of the two constraints leads to

$$H = W \left[\frac{\partial c_1}{\partial \boldsymbol{\theta}} , \frac{\partial c_2}{\partial \boldsymbol{\theta}} \right] \tag{16.29}$$

with

$$\frac{\partial c_1}{\partial \boldsymbol{\theta}} = \mathbf{1} , \quad \frac{\partial c_2}{\partial \boldsymbol{\theta}} = \boldsymbol{\theta}^0 = [z_i^0] \quad \text{and} \quad W = \mathrm{Diag}([w_i]) . \tag{16.30}$$

Observe, we asumed the approximate value s^0 for the scaled rotation of the point set in (16.16) is 1.

As an example, if all weights are zero except for $w_1 = w_2 = 1$, then the first two points are not changed compared to the approximate values, i.e., fix the gauge, and the resulting constraint matrix is

$$H^W = W[\mathbf{1}, \boldsymbol{\theta}^0] , \tag{16.31}$$

only having non-zeros in the first two rows, namely being

$$\begin{bmatrix} 1 & z_1^0 \\ 1 & z_2^0 \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \end{bmatrix} . \tag{16.32}$$

16.3 Using the gauge within the estimation

The gauge constraints can easily be intorduces using a Gauss–Markov model with constraints. The covariance matrix of the parameters $\boldsymbol{\theta}$ can be derived from

$$\begin{bmatrix} \Sigma_{\hat{\theta}\hat{\theta}} & . \\ . & . \end{bmatrix} = \begin{bmatrix} A^\mathsf{T}\Sigma_{yy}^{-1}A & WH \\ H^\mathsf{T}W & 0 \end{bmatrix} \tag{16.33}$$

(Förstner and Wrobel, 2016, Sect. 4.3). Observe, the transposition includes conjugation of the complex elements.

In the Cinderella animation we show the circular standard ellipses of the points $\hat{z}_i$ based on the standard deviations

$$\sigma_{x_i} = \sigma_{y_i} = \frac{\sigma_{\theta_i}}{\sqrt{2}} = \sqrt{\frac{\left(\Sigma_{\hat{\theta}\hat{\theta}}\right)_{ii}}{2}} . \tag{16.34}$$

This is the way the script in the Cinderella animation is realized.

16.4 Changing the gauge

We could also start with an arbitrary choice of the gauge, and afterwards change the gauge using a gauge or S -transformation (standing for similarity transformation), (Förstner and Wrobel, 2016, Sect. 4.5.3)

We can chang the gauge by specifying a weightmatrix $W = \mathrm{Diag}([w_i])$ and projecting the estimated parameters to a subspace of size $I - 2$ (referring to complex variables)

$$S^W = I - H(H^\mathsf{T}WH)^{-1}H^\mathsf{T}W \tag{16.35}$$

namely

$$\Delta\boldsymbol{\theta}^W = S^W \, \Delta\boldsymbol{\theta} . \tag{16.36}$$

where the transposition includes conjugation of the elements of the matrix H. The projection fixes the gauge by enforcing the constraints

$$c_1 = \sum_i w_i \, (\hat{z}_i - \overline{z}) = 0 \quad \text{and} \quad c_2 = \sum_i w_i (s \, (\hat{z}_i - \overline{z}) + t - z_i^0) = 0 , \tag{16.37}$$

where the weights need not be the same as used for deriving the intial covariance matrix.

16.5 The uncertainty of arbitrary local scaled rotations

Local scaled rotations are invariant to the choice of the gauge/datum. Therefore, also their uncertainty is an invariant quantity. Using variance propagation we are able to derive the variance of some arbitrary local scaled rotation $\underline{\Pi}_{jik}$ using the Jacobian in (16.25). We have

$$\mathbb{V}(\underline{\Pi}_{jik}) = \underbrace{\boldsymbol{a}_{i0}^\mathsf{T}}_{1\times 3} \underbrace{[\Sigma_{\theta\theta}]_{jik}}_{3\times 3} \underbrace{\boldsymbol{a}_{i0}}_{3\times 1} \tag{16.38}$$

where the 3×3 matrix is the submatrix of $\Sigma_{\theta\theta}$ referring to the three points (jik). In the Cinderella animation we report the standard deviations of the corresponding logarithmic scale ratio and the angle:

$$\sigma_{\alpha_{jik}} = \frac{\sigma_{r_{jik}}}{r_{jik}} = \frac{\sigma_{\Pi_{jik}}}{\sqrt{2}} = \sqrt{\frac{\mathbb{V}(\underline{\Pi}_{jik})}{2}}\,. \tag{16.39}$$

17 Markov-Random Fields and Geodetic Networks

Geodetic networks are specific Markov random fields. This is shown by relating the usually sparse graph of observations to the information matrix, the normal equation matrix, which encodes the conditional independencies of the parameters, namely the coordinates.

17.1 Preface

The note (2009) shows that geodetic networks are specific Markov random fields. This is shown by relating the usually sparse graph of observations to the information matrix, which usually is referred to as normal equation matrix and which encodes the conditional independencies of the parameters. The results of this note are partly use in Förstner (2013).

17.2 The Problem

Geodetic networks inherently lead to sparse observation and normal equations. The network structure explicitly is reflected in the non-zero-structure which can be derived from the adjacency matrix of the observation graph.

Markov-Random Fields rely on a sparse conditional independence structure. This is given by the set of maximal cliques which in many cases only contain a few, sometimes only two, nodes in a graph.

This note shows the equivalence of both models. The equivalence also transfers to certain methods to find optimal solutions, namely the iterative method of iterative conditional modes and the Gauss-Seidel-iteration method.

17.3 Markov-Random-Fields

17.3.1 General form

Given is a graph with nodes and undirected edges. The nodes $\mathcal{X} = \{x_n, n = 1, ..., N\}$ represent random variables $\underline{x}_n$. In general, we do not distinguish between the nodes themselves and the random variables they stand for. In case we do, the nodes are denoted by the indices of the random variables, so we also could write $\mathcal{X} = \{n, n = 1, ..., N\}$ where n stands for the node representing the variable $\underline{x}_n$.

No constraint is imposed on the type of random variable. They may be continuous, discrete or mixed.

We denote sets of nodes or random variables with calligraphic letters. Sometimes we prefer to collect scalar random variables in a vector.

The edges denote some weak constraints to be specified below. The graph structure is meant to show conditional independencies. Let $\mathcal{N}_x$ be the neighbours of a node x. In case the graph represents a Markov-random field (MRF) we have the following Markov-relation

$$p(x|\mathcal{X} \setminus x) = p(x|\mathcal{N}_x) \tag{17.1}$$

I. e. the probability of x given the values of all other nodes $\mathcal{X} \setminus x$ is equal to the probability of x given its neighbours $\mathcal{N}_x$.

The probability $p(\mathcal{X})$ of all nodes can be now be related to functions $\psi_C(\mathcal{X}_C)$ of the values of the maximal cliques $\mathcal{X}_C$ by

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{x_C} \psi_C(\mathcal{X}_c) \tag{17.2}$$

This requires some explanation:

- The maximal cliques $\mathcal{X}_C$ are given by the application. One may restrict to two-cliques even in case three nodes are pairwise neighbours. All k-cliques $\mathcal{C}_k$ with k nodes, contain smaller k'-cliques $\mathcal{C}_{k'}$, with $k' < k$. The relation between these smaller cliques need be taken into account, as their relation can be captured in the function ψ_C.
- The functions ψ_C are called potential functions. They are assumed to be positive. Therefore, it may be useful to write them as exponentials

  $$\psi_C(\boldsymbol{X}_C) = \mathrm{e}^{-U_C(\mathcal{X}_C)} \tag{17.3}$$

  The functions $U_C(\mathcal{X}_C) = -\log \psi_C(\mathcal{X}_C)$ are called energy functions.
  The potential functions may contain some additional unknown parameters, Θ, parametrizing the stochastic properties.
  As the probability of the complete set of nodes should be maximal when searching for parameters hidden in the potential functions, the potential functions should be chosen such that likely parameter values for x_n lead to large values of $\psi_C(\mathcal{X}_C)$ or small values of $U_C(\mathcal{X}_C)$.
- The constant Z, sometimes called the partition function, is used to normalize the product such that the integral over $p(\mathcal{X})$ over *all* possible states of $\mathcal{X}$ is 1. This usually hinders to determine probabilities. However, often one may use the probabilities as preference measure, then the knowledge of Z is not necessary.
- Maximizing the probability $p(\mathcal{X})$ is equivalent to minimizing

  $$\Omega = -\log p(\mathcal{X}) = \sum_{x_C} U_C(\mathcal{X}_C) + \log Z \tag{17.4}$$

  In case the partition function Z is constant, though unknown, this is the sum of the individual energy functions.

17.3.2 Solution techniques

17.3.2.1 Iterative Conditional Mode

A simple solution technique is the so-called iterative conditional mode method. Here all nodes $\mathcal{X} \setminus x$ except one x are assumed to be known and one determines

$$x^{(\nu+1)} = \text{argmax}_x p(x|\mathcal{N}_x^{(\nu)}) \tag{17.5}$$

using the relation (17.1).

The sequence of visiting the nodes can be chosen randomly or in a prespecified order. A prespecified order may guarantee that each node is really visited regularly. Observe that the total number of iterations will in general be several times the number $|\mathcal{X}|$ of nodes in order to guarantee convergence. The solution will depend on the initial values, thus in general yields a local maximum.

The situation may be improved by simultaneously updating multiple nodes, which are not mutual neighbours. In case of a gridded structure with a 4-neighbourhood on may update each second node in each row, taking the odd nodes in the odd rows and the even nodes in the even rows in a first run, and the other nodes, cf. fig. 17.1. Thus one needs to colour the graph, such than no two neighbouring nodes have the same colour and then repeat the update of the nodes following the sequence of the colours. Obviously, the update of all nodes of the same colour can be done in parallel.

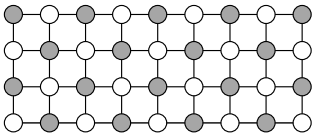


Figure 17.1: Colouring of a graph with 4-neighbourhood

17.3.2.2 Min-Cut-Algorithm

In case the nodes are binary nodes, we maximally have three-cliques and the optimization function fulfils certain regularity constraints one can solve the problem *rigorously* using the so-called min-cut-algorithm[1]. The idea is to extend the graph by two nodes, a source and a sink node connected to all nodes of the given graph, and find the cheapest set of edges, a cut of the graph, to interrupt the flow from the source to the sink.

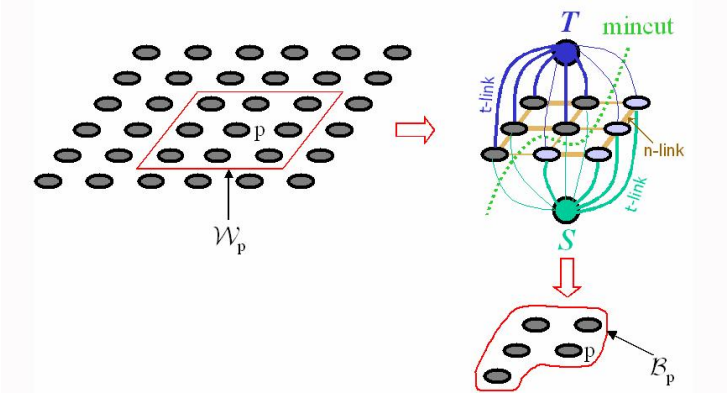


Figure 17.2: Tree for determining minimum cut

[1] cf. Y. Boykov and V. Kolmogoroff (2004): An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, In IEEE Transactions on PAMI, Vol. 26, No. 9, pp. 1124-1137, Sept. 2004

17.3.2.3 Monte Carlo Techniques

In more general situations, especially in case no good approximate values are available, on need to use Monte-Carlo-techniques. Their idea is to change the values at the nodes in a sequence, where the new configuration, i. e. set of values $\mathcal{X}^{(\nu+1)}$, depends randomly on the previous configuration $\mathcal{X}^{(\nu)}$. With the sequence of iterations, the difference between successive states are controlled to get smaller at the same time guaranteeing to reach a probable configuration. One of the techniques is the *simulated annealing*. Here the generation of crystals is simulated, by starting with a liquid and slowly cooling the material.

17.4 The probability distribution of geodetic networks

Geodetic networks usually lead to coordinates of sets of points. Their uncertainty is represented by a covariance matrix, implicitly assuming a multivariate normal distribution. In the following we analyse how independence between multivariate Gaussian variables $\boldsymbol{x}$ is encoded.

17.4.1 Independence, correlation and conditional independence

The uncertainty is represented by their covariance matrix Σ_{xx} or their precision matrix

$$P_{xx} = \Sigma_{xx}^{-1} \tag{17.6}$$

This precision matrix is identical to the weight matrix in case $\sigma_0^2 = 1$. Thus we may write

$$\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx}) = \mathcal{N}(\boldsymbol{\mu}_x, P_{xx}^{-1}) \tag{17.7}$$

It will turn out that it might be beneficial to use the precision matrix P_{xx} instead of the covariance matrix Σ_{xx}.

In case we split the complete vector $\boldsymbol{x}$ in two parts we obtain the complete description

$$\begin{bmatrix} \underline{\boldsymbol{y}} \\ \underline{\boldsymbol{z}} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu}_y \\ \boldsymbol{\mu}_z \end{bmatrix}, \begin{bmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_{zz} \end{bmatrix}\right) = N\left(\begin{bmatrix} \boldsymbol{\mu}_y \\ \boldsymbol{\mu}_z \end{bmatrix}, \begin{bmatrix} P_{yy} & P_{yz} \\ P_{zy} & P_{zz} \end{bmatrix}^{-1}\right) \tag{17.8}$$

The relation between the submatrices of Σ_{xx} and P_{xx} is well known:

$$P_{xx} = \begin{bmatrix} \overline{\Sigma}_{yy}^{-1} & -\overline{\Sigma}_{yy}^{-1}\Sigma_{yz}\Sigma_{zz}^{-1} \\ -\Sigma_{zz}^{-1}\Sigma_{zy}\overline{\Sigma}_{yy}^{-1} & \Sigma_{zz}^{-1} + \Sigma_{zz}^{-1}\Sigma_{zy}\overline{\Sigma}_{yy}^{-1}\Sigma_{yz}\Sigma_{zz}^{-1} \end{bmatrix} \tag{17.9}$$

with the Schur-complement

$$\overline{\Sigma}_{yy} = \Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{zy} \tag{17.10}$$

Inversely we obtain

$$\Sigma_{xx} = \begin{bmatrix} \overline{W}_{yy}^{-1} & -\overline{W}_{yy}^{-1}P_{yz}P_{zz}^{-1} \\ -P_{zz}^{-1}P_{zy}\overline{W}_{yy}^{-1} & P_{zz}^{-1} + P_{zz}^{-1}P_{zy}\overline{W}_{yy}^{-1}P_{yz}P_{zz}^{-1} \end{bmatrix} \tag{17.11}$$

with the Schur-complement

$$\overline{W}_{yy} = P_{yy} - P_{yz}P_{zz}^{-1}P_{zy} \tag{17.12}$$

Marginal distribution. The marginal distribution of $\underline{\boldsymbol{y}}$ is

$$\underline{\boldsymbol{y}} \sim N(\boldsymbol{\mu}_y, \Sigma_{yy}) \tag{17.13}$$

Thus the uncertainty of the *marginal distribution* is obtained by selecting the appropriate rows and columns in the *covariance matrix*.

Conditional distribution. We also need the distribution of $\underline{\boldsymbol{y}}$ in case $\boldsymbol{z}$ is given. This is

$$\underline{\boldsymbol{y}}|\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_{y|z}, \Sigma_{yy|z}) \tag{17.14}$$

with

$$\boldsymbol{\mu}_{y|z} = \boldsymbol{\mu}_y + \Sigma_{yz}\Sigma_{zz}^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_z) \qquad \Sigma_{yy|z} = \Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{zy} \tag{17.15}$$

or using the precision matrices

$$\boldsymbol{\mu}_{y|z} = \boldsymbol{\mu}_y - P_{yy}^{-1}P_{yz}(\boldsymbol{z} - \boldsymbol{\mu}_z) \qquad \Sigma_{yy|z} = P_{yy}^{-1} \tag{17.16}$$

The proof can be done easily by expanding the exponent of the Gaussian of $\underline{\boldsymbol{x}}$, cf.[2].

Thus the uncertainty of the *conditional distribution* is obtained by selecting the appropriate rows and columns in the *precision matrix and inversion*.

Independence. Two variates $\underline{\boldsymbol{y}}_j$ and $\underline{\boldsymbol{y}}_j$ are stochastically independent, in case they are uncorrelated, thus in case $\Sigma_{y_i y_j} = 0$:

$$\boldsymbol{y}_i \perp\!\!\!\perp \boldsymbol{y}_j \Leftrightarrow \Sigma_{y_i y_j} = 0 \tag{17.17}$$

Conditional independence. Two variates $\underline{\boldsymbol{y}}_i$ and $\underline{\boldsymbol{y}}_j$ are conditionally independent given $\boldsymbol{z}$, in case they are conditionally uncorrelated, thus in case $\Sigma_{y_i y_j|z} = 0$. Thus, using eq. (17.16)

$$\boldsymbol{y}_i \perp\!\!\!\perp \boldsymbol{y}_j|\boldsymbol{z} \Leftrightarrow \Sigma_{y_i y_j|z} = \left(P^{-1}\right)_{y_i y_j} = 0 \tag{17.18}$$

This gives a method to determine conditional independencies, as P_{yy} is obtained by cancelling the rows and columns in P_{xx} belonging to $\boldsymbol{z}$. In case $\boldsymbol{z}$ is chosen such that P_{yy} is block diagonal,

$$\boldsymbol{y}_1 \perp\!\!\!\perp \boldsymbol{y}_2|\boldsymbol{z} \quad \Leftrightarrow \quad \Sigma_{yy|z} = \begin{bmatrix} P_{y_1 y_1} & 0 \\ 0 & P_{y_2 y_2} \end{bmatrix}^{-1} \tag{17.19}$$

the different blocks will also be uncorrelated: Then the two subsets $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ of parameters are conditionally independent w. r. t. $\boldsymbol{z}$.

17.4.2 Independencies in geodetic networks

The precision of geodetic networks usually is represented using the covariance Σ_{kk} of their coordinates $\boldsymbol{k}$, implicitly assuming their distribution is a multi-variate Gaussian. Storing the full covariance matrix may be prohibitive, in case the network is quite large, with, say, more than 100000 points. The reason is simple, the integration of all measurements within a simultaneous estimation process leads to a full covariance matrix in general.

The integration of different sets of measurements usually is done with the help of the normal equations $N\boldsymbol{k} = \boldsymbol{h}$ as

$$N = \sum_t N_t \qquad \boldsymbol{h} = \sum_t \boldsymbol{h}_t \tag{17.20}$$

Moreover, the normal equations usually are sparse.

This leads us to the following observations

- Coordinates in geodetic networks usually are correlated, thus not statistically independent. This holds even for points which are quite far apart, the correlations may be small, but they do not vanish.
- Coordinates in geodetic networks show a sparse precision structure, encoded in the normal equation matrix[3] $N = \Sigma_{kk}^{-1}$. The normal equations usually are quite sparse. Zeros in the precision matrix N go along with conditional independence. We will come back to this below.

[2] this is: $-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_x)^\mathsf{T} P_{xx}(\boldsymbol{x} - \boldsymbol{\mu}_x) = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_y)^\mathsf{T} P_{yy}(\boldsymbol{y} - \boldsymbol{\mu}_y) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_y)^\mathsf{T} P_{yz}(\boldsymbol{z} - \boldsymbol{\mu}_z) - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu}_z)^\mathsf{T} P_{zy}(\boldsymbol{y} - \boldsymbol{\mu}_y) - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu}_z)^\mathsf{T} P_{zz}(\boldsymbol{z} - \boldsymbol{\mu}_z)$ fixing all values for $\boldsymbol{z}$ and rearranging the terms to obtain a quadratic expression $(\boldsymbol{y} - \boldsymbol{a})^\mathsf{T} P_{yy}(\boldsymbol{y} - \boldsymbol{a})$ finding $\boldsymbol{a}$ to be $\boldsymbol{\mu}_{y|z}$

[3] assuming the variance factor to be $\sigma_0^2 = 1$.

17.5 An illustrative example

17.5.1 The given graph

Let the small graph with $N = 4$ nodes $\mathcal{X} = \{1, 2, 3, 4\}$ be given. Let us further assume five edges $\mathcal{E} = \{(1,2), (2,3), (3,4), (1,4), (2,4)\}$ connect these nodes. Then we have 4 one-cliques $\mathcal{C}_1 = \{1, 2, 3, 4\}$, five two-cliques, $\mathcal{C}_2 = \mathcal{E}$ and two three-cliques $\mathcal{C}_3 = \{(1,2,4), (2,3,4)\}$

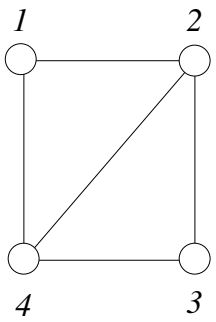


Figure 17.3: A small Markov-Random-Field

In case we restrict ourselves to two-cliques as maximal cliques the general form of the joint probability of the four nodes can be written as

$$\begin{aligned} p(\mathcal{X}) &= p(x_1, x_2, x_3, x_4) && (17.21)\\ &= \frac{1}{Z} \prod_{(ij)\in\mathcal{E}} \psi_{ij}(x_i, x_j) && (17.22)\\ &= \frac{1}{Z} \psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)\psi_{34}(x_3, x_4)\psi_{14}(x_1, x_4)\psi_{14}(x_2, x_4) && (17.23) \end{aligned}$$

as we only have 5 two-cliques, the arcs.

In case we allow for three-cliques as maximal cliques we obtain the general form of the joint probability

$$\begin{aligned} p(\mathcal{X}) &= \frac{1}{Z} \prod_{(ijk)\in\mathcal{C}_3} \psi_{ijk}(x_i, x_j, x_k)\\ &= \frac{1}{Z} \psi_{124}(x_1, x_2, x_4)\psi_{234}(x_2, x_3, x_4) && (17.24) \end{aligned}$$

17.5.1.1 Example 1a: Potential function for a levelling network

Let the graph represent a levelling network. The nodes are points with height x_i. We assume we observe the five height differences, leading to values d_{ij}. In addition, we have some information about the heights of the points 1 and 3. Then we can model the situation as a MRF.

The x_i are continuous scalar variables. We further assume the following potential

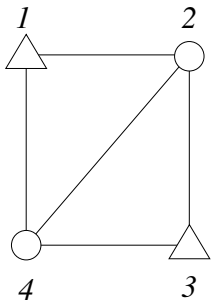


Figure 17.4: A levelling network as MRF

functions

$$\psi_{12} = \mathrm{e}^{-\frac{1}{2}(x_1 - x_2 - d_{12})^2 w_{12}} \mathrm{e}^{-\frac{1}{2}(x_1 - d_1)^2 w_1} \tag{17.25}$$

$$\psi_{23} = \mathrm{e}^{-\frac{1}{2}(x_2 - x_3 - d_{23})^2 w_{23}} \tag{17.26}$$

$$\psi_{34} = \mathrm{e}^{-\frac{1}{2}(x_3 - x_4 - d_{34})^2 w_{34}} \mathrm{e}^{-\frac{1}{2}(x_3 - d_3)^2 w_1} \tag{17.27}$$

$$\psi_{14} = \mathrm{e}^{-\frac{1}{2}(x_1 - x_4 - d_{14})^2 w_{14}} \tag{17.28}$$

$$\psi_{14} = \mathrm{e}^{-\frac{1}{2}(x_2 - x_4 - d_{24})^2 w_{24}} \tag{17.29}$$

where d_1 and the d_{ij} are given numbers, namely the given heights and the observed height differences. The potentials ψ_{ij} are large in case the difference $x_i - x_j$ are close to d_{ij}, deviations are assumed to be normally distributed. The given values $w_{ij} := p_{ij}$ may be used to weight the differences. The first and the third potential in addition prefers values for x_1 close to d_1 and values x_3 close to d_3.

The probability density of $\mathcal{X}$ thus reads as

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{i=1}^{4} \mathrm{e}^{-\frac{1}{2}(x_i - d_i)^2 w_i} \prod_{(ij)\in\mathcal{E}} \mathrm{e}^{-\frac{1}{2}(x_i - x_j - d_{ij})^2 w_{ij}} \tag{17.30}$$

with $p_2 = p_4 = 0$.

The energy function thus reads as

$$\Omega = \frac{1}{2} \left[\sum_{i=1}^{4} (x_i - d_i)^2 w_i + \sum_{(ij)\in\mathcal{E}} (x_i - x_j - d_{ij})^2 w_{ij} \right] \tag{17.31}$$

17.5.1.2 Example 1b: Potential function for a distance network

We now assume the graph represents a geodetic network in the plane. The nodes therefore represent points with coordinates $x_i = [u_i, v_i]^\mathsf{T}$. We additionally assume to observe five distances d_{ij} between the points.

The x_i again are continuous variables, namely 2-vectors with components $x_i = [u_i, v_i]^\mathsf{T}$.

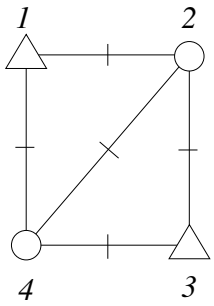


Figure 17.5: A planar network with measured distances as MRF

We further assume the following potential functions

$$\psi_{12} = \mathrm{e}^{-\frac{1}{2}(|x_1 - x_2| - d_{12})^2 w_{12}} \mathrm{e}^{-\frac{1}{2}|x_1 - d_1|^2 w_1} \tag{17.32}$$

$$\psi_{23} = \mathrm{e}^{-\frac{1}{2}(|x_2 - x_3| - d_{23})^2 w_{23}} \tag{17.33}$$

$$\psi_{34} = \mathrm{e}^{-\frac{1}{2}(|x_3 - x_4| - d_{34})^2 w_{34}} \mathrm{e}^{-\frac{1}{2}|x_3 - d_3|^2 w_1} \tag{17.34}$$

$$\psi_{14} = \mathrm{e}^{-\frac{1}{2}(|x_1 - x_4| - d_{14})^2 w_{14}} \tag{17.35}$$

$$\psi_{24} = \mathrm{e}^{-\frac{1}{2}(|x_2 - x_4| - d_{24})^2 w_{24}} \tag{17.36}$$

where the d_i are given 2-vectors and the d_{ij} are given positive numbers, representing the given positions and the observed distances. The potentials ψ_{ij} are large in case the norms $|x_i - x_j|$ of the differences are close to d_{ij}. The given values p_{ij} may be used to weight the differences. The first and the third potential in addition prefers values for x_1 close to d_1 and x_3 close to d_3.

The probability density of $\mathcal{X}$ thus reads as

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{i=}^{4} \mathrm{e}^{-\frac{1}{2}|x_i - d_i|^2 w_i} \prod_{(ij)\in\mathcal{E}} \mathrm{e}^{-\frac{1}{2}(|x_i - x_j| - d_{ij})^2 w_{ij}} \tag{17.37}$$

The energy function thus reads as

$$\Omega = \frac{1}{2} \left[\sum_{i=1}^{4} |x_i - d_i|^2 w_i + \sum_{(ij)\in\mathcal{E}} (|x_i - x_j| - d_{ij})^2 w_{ij} \right] \tag{17.38}$$

17.5.1.3 Example 1c: Potential functions for an angle network

We now again assume the graph to represent a geodetic network in the plane. However, we now observe angles d_{ijk} at point j between points i and k. Then we need to take into account potentials of 3-cliques. The angles constrain the form of triangles, thus all three points of such a triangle need to be pairwise neighbours. In case we want to leave the graph unchanged we therefore can introduce angle measurements in the triangles $(1, 2, 4)$ and $(2, 3, 4)$. In case we also would observe angle d_{123} we would need to introduce an edge between 1 and 3.

The nodes again represent two-vectors. The potentials of the two 3-cliques are assumed

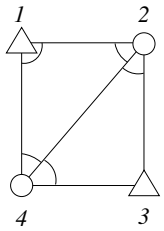


Figure 17.6: A planar network with measured angles as MRF

to be

$$\begin{aligned}\psi_{124}(x_1,x_2,x_4) &= \mathrm{e}^{-\frac{1}{2}\left(|a_{421}-d_{421}|^2p_{421}+|a_{142}-d_{142}|^2p_{142}+|a_{214}-d_{214}|^2p_{214}+|x_1-d_1|^2p_1\right)}\\ \psi_{234}(x_2,x_3,x_4) &= \mathrm{e}^{-\frac{1}{2}\left(|a_{243}-d_{243}|^2p_{243}+|a_{324}-d_{324}|^2p_{324}+|x_3-d_3|^2p_3\right)}\end{aligned} \tag{17.39}$$

where we have the nonlinear function

$$a_{i,j,k}(x_i,x_j,x_k) = r_{jk}(x_j,x_k) - r_{ji}(x_j,x_i) \qquad r_{jk}(x_j,x_k) = \operatorname{atan2}(v_k-v_j, u_k-u_j) \tag{17.40}$$

being the angle at j between the directions to i and k.

Obviously, the two potentials represent 5 angle measurements and two point measurements, being unbiased observations of the angles and the point coordinates with a precision reflected in the weights p_{ijk} and p_i.

17.5.1.4 Example 2: Potential function for a classification network

Now, let us assume the x_i are discrete and binary variables $x_i \in \{-1,1\}$. We further assume the following potential functions

$$\psi_{12} = \mathrm{e}^{\beta_{12}x_1x_2}\mathrm{e}^{\alpha_1x_1d_1+\alpha_2x_2d_2} \tag{17.41}$$

$$\psi_{23} = \mathrm{e}^{\beta_{23}x_2x_3} \tag{17.42}$$

$$\psi_{34} = \mathrm{e}^{\beta_{34}x_3x_4}\mathrm{e}^{\alpha_3x_3d_3-\alpha_4x_4d_4} \tag{17.43}$$

$$\psi_{14} = \mathrm{e}^{\beta_{14}x_1x_4} \tag{17.44}$$

In case the given values $\beta_{ij} > 0$ the potential functions favour the same value for neighbouring nodes, as then the product $x_ix_j = 1$, otherwise they favour different values. The degree of preference is coded in the absolute value of the β_{ij}'s. In case the given values $\alpha_i > 0$ the first and the third potential function favour values of $x_i = d_i$, where $d_i \in \{-1,1\}$ are given values. These preferences are weighted with the given values α_i.

The example can be interpreted as follows. Assume, we have four neighbouring parcels, which either contain a building or not. Let us assume we may derive the land cover of the parcels from a satellite, which yields an uncertain classification of the parcels.

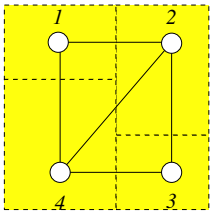


Figure 17.7: A set of neighbouring parcels as MRF

This knowledge may be represented by

$$p(\mathcal{X}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{X})p(\mathcal{X}) \tag{17.45}$$

where $p(\mathcal{X})$ represents the a priori knowledge about the classes and $p(\mathcal{D}|\mathcal{X})$ the observation process of the classification procedure.

Let us assume the we know from previous analysis', that it is likely that two neighbouring parcels usually have the same or different land cover. This knowledge may be encoded as

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{(ij)\in\mathcal{E}} \mathrm{e}^{\beta_{ij} x_i x_j} \tag{17.46}$$

E. g. in case $\beta_{ij} = 1$ we have $\psi_{ij} = \mathrm{e} \approx 2.72$ in case the classes of two neighbouring sites are equal and $\psi_{ij} = 1/\mathrm{e} \approx 0.368$, i. e. in case the classes of two neighbouring sites are equal the probability for the situation is $\mathrm{e}^2 \approx 7.4$ more probable than if the two sites have different classes, all other nodes having the same label.

The likelihood term may be represented as

$$p(d_i|x_i) = \mathrm{e}^{\alpha_i d_i x_i} \tag{17.47}$$

Again, in case $\alpha_i = 1$ the probability of the complete situation is $\mathrm{e}^2 \approx 7.4$ times more probable in case $d_i = x_i$ than if $d_i \neq x_i$.

Then the complete knowledge may be represented as

$$p(\mathcal{X}) = \frac{1}{Z} \prod_n \mathrm{e}^{\alpha_n d_n x_n} \prod_{(ij)\in\mathcal{E}} \mathrm{e}^{\beta_{ij} x_i x_j} \tag{17.48}$$

Thus, in general, the probability density for $\mathcal{X}$ reads as

$$p(\mathcal{X}) = \frac{1}{Z} \exp\left(\sum_{n=1}^{4} \alpha_n d_n x_n + \sum_{(ij)\in\mathcal{E}} \beta_{ij} x_i x_j \right) \tag{17.49}$$

whereas the energy function reads as

$$Q = -\left(\sum_{n=1}^{4} \alpha_n d_n x_n + \sum_{(ij)\in\mathcal{E}} \beta_{ij} x_i x_j \right) \tag{17.50}$$

In our special case we have $\alpha_2 = \alpha_4 = 0$, i. e. any observation of nodes x_2 and x_4 have no influence.

17.5.2 Solutions for the example

17.5.2.1 Solutions for the continuous variables

In order to find an optimal set of values $\mathcal{X}$ we need to take the partial derivatives of $p(\mathcal{X})$ with respect to the unknown values x_i or equivalently the partial derivatives of $-\log p(\mathcal{X})$ and solve the four equations

$$\frac{\partial(-\log p(\mathcal{X}))}{\partial x_n} = 0 \qquad n = 1, ..., 4 \tag{17.51}$$

These derivatives lead to 4 equations, which in general are nonlinear. We discuss the first example, since the resulting equations are linear.

17.5.2.2 Numerical solution for the Levelling network

In this case we have the partials

$$\begin{aligned}
\frac{\partial(-\log p(\mathcal{X}))}{\partial x_1} &= 0 = (x_1 - x_2 - d_{12})w_{12} + (x_1 - d_1)w_1 + (x_1 - x_4 - d_{14})w_{14} \\
\frac{\partial(-\log p(\mathcal{X}))}{\partial x_2} &= 0 = -(x_1 - x_2 - d_{12})w_{12} + (x_2 - x_3 - d_{23})w_{23} + (x_2 - x_4 - d_{24})w_{24} \\
\frac{\partial(-\log p(\mathcal{X}))}{\partial x_3} &= 0 = -(x_2 - x_3 - d_{23})w_{23} + (x_3 - x_4 - d_{34})w_{34} + (x_3 - d_3)w_3 \\
\frac{\partial(-\log p(\mathcal{X}))}{\partial x_4} &= 0 = -(x_3 - x_4 - d_{34})w_{34} - (x_1 - x_4 - d_{14})w_{14} - (x_2 - x_4 - d_{24})w_{24}
\end{aligned}$$

which can be written as the following equation system

$$N\boldsymbol{x} = \boldsymbol{h} \tag{17.52}$$

with

$$N = \begin{bmatrix} p_{12} + p_{14} + p_1 & -p_{12} & 0 & -p_{14} \\ -p_{12} & p_{12} + p_{23} + p_{24} & -p_{23} & -p_{24} \\ 0 & -p_{23} & p_{23} + p_{34} + p_3 & -p_{34} \\ -p_{14} & -p_{24} & -p_{34} & p_{34} + p_{14} + p_{24} \end{bmatrix} \tag{17.53}$$

$$\boldsymbol{h} = \begin{bmatrix} p_{12}d_{12} + p_{14}d_{14} + p_1 d_1 \\ -p_{12}d_{12} + p_{23}d_{23} + p_{24}d_{24} \\ -p_{23}d_{23} + p_{34}d_{34} + p_3 d_3 \\ -p_{34}d_{34} - p_{14}d_{14} - p_{24}d_{24} \end{bmatrix} \tag{17.54}$$

Observe, in case we fix the values of nodes 2 and 4, then the normal equation system reduces to a 2×2-system for the variables 1 and 3: As the off-diagonal element $(1, 3)$ in N is zero, the two variables x_1 and x_3 are conditionally independent, given x_2 and x_4:

$$x_1 \perp\!\!\!\perp x_3 | x_2, x_4 \tag{17.55}$$

This is directly visible from the graph.

Moreover, here the ICM-algorithm leads to the following iterative scheme, in case the weights p_i and p_{ij} are all assumed to be 1:

$$x_1^{(\nu+1)} = \frac{1}{3}(h_1 + x_2^{(\nu)} + x_4^{(\nu)}) \tag{17.56}$$

$$x_2^{(\nu+1)} = \frac{1}{3}(h_2 + x_1^{(\nu+1)} + x_3^{(\nu)} + x_4^{(\nu)}) \tag{17.57}$$

$$x_3^{(\nu+1)} = \frac{1}{3}(h_3 + x_2^{(\nu+1)} + x_4^{(\nu)}) \tag{17.58}$$

$$x_4^{(\nu+1)} = \frac{1}{3}(h_4 + x_1^{(\nu+1)} + x_3^{(\nu+1)} + x_2^{(\nu+1)}) \tag{17.59}$$

This exactly is what is called the Gauss-Seidel-Iteration scheme (see http://de.wikipedia.org/wiki/Gau%C3%9F-Seidel-Verfahren), known from numerical mathematics for iteratively solving equation systems.

Part III

Technical Notes on Estimation of Geometric Entities

18 Motions and their Uncertainty

We address the ambiguity of representing uncertain motions. We analyse the relation between an exponential representation with a homogeneous 4x4 matrix and the representation with a rotation matrix, also represented exponentially, and a translation vector. The rotation parts turns out to be identical, while the translation parts differ, why a transparent documentation of the motion representation is necessary. As a sideline, the note addresses the inversion, the concatenation, and the difference between uncertain rotations and uncertain motions.

18.1 Preface

The note initially (2009) was motivated by the need to concatenate uncertain motions. Later, in 2017 an extension was motivated by the search for an error in a program for estimating the motion between two point clouds based on corresponding planar regions. The error turned out to be a conceptual one: the generation of the test data and the check of the estimated motions was inconsistent, since one used the exponential form of a motion, while the other used the exponential form of a rotation and the translation. For a detailed discussion see Solà et al. (2018).

18.2 Motivation

This note is motivated by a problem when handling uncertain 3D motions or poses: The two classical representations, the one what we call the *exponential* representation and the other what we call *partially exponential* representation, may both be used for estimating motions or poses, but lead to different covariance matrices of the translation component. The note aims at clarifying the mutual relations between the different representations.

The *exponential representation* of an uncertain motion with mean rotation R and mean translation $\boldsymbol{Z}$, exploits the Lie group structure of the noise component of the motion using what is called a twist vector $\underline{\boldsymbol{s}}$, which contains the noise components $\underline{\boldsymbol{r}}$ and $\underline{\boldsymbol{t}}$ for rotation and translation,[1] in the form

$$^{s}\underline{\mathsf{M}} = \exp(\mathsf{A}(\underline{\boldsymbol{s}}))\ \mathsf{M}\,, \tag{18.1}$$

with

$$\mathsf{M} = \begin{bmatrix} R & \boldsymbol{Z} \\ \boldsymbol{0}^{\mathsf{T}} & 1 \end{bmatrix}\,,\quad \boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{t} \end{bmatrix}\,,\quad \mathsf{A}(\boldsymbol{s}) = \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \boldsymbol{0}^{\mathsf{T}} & 0 \end{bmatrix}\,, \tag{18.2}$$

and

$$S(\boldsymbol{r}) = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}\,. \tag{18.3}$$

The matrix $\mathsf{A}(\boldsymbol{s})$ is close to zero, such that the motion matrix $\exp(\mathsf{A}(\underline{\boldsymbol{s}}))$ is close to I_4.

The *partially exponential representation* directly integrates the noise components $\underline{\boldsymbol{\rho}}$ and $\underline{\boldsymbol{\tau}}$ for a small rotation and translation

$$^{\zeta}\underline{\mathsf{M}} = \begin{bmatrix} \exp(S(\underline{\boldsymbol{\rho}}))R & \boldsymbol{Z} + \underline{\boldsymbol{\tau}} \\ \boldsymbol{0}^{\mathsf{T}} & 1 \end{bmatrix}\,, \tag{18.4}$$

into M thus only applies the exponential map to the noise component of the rotation. Again, since $\underline{\boldsymbol{\rho}}$ is small, the rotation matrix $\exp(S(\underline{\boldsymbol{\rho}}))$ is close to I_3.

The following example shows the effect of the different representations. Given are 100 random samples of an uncertain motion together with the true motion. From this sample we may obtain two covariance matrices Σ_1 and Σ_2 with the following vectors of standard deviations for the rotational and the translational component:

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} 0.5109 \\ 0.4803 \\ 0.2760 \\ 0.4696 \\ 0.3324 \\ 0.3591 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\sigma}_2 = \begin{bmatrix} 0.5109 \\ 0.4803 \\ 0.2760 \\ 0.6125 \\ 0.6189 \\ 0.7433 \end{bmatrix}\,. \tag{18.5}$$

The rotation parameters have the same standard deviations, while the standard deviations of the translational component significantly differ. Without further information we cannot judge, which covariance matrix is the correct one. We would prefer the first one,

[1] Random variables are underscored.

since it shows smaller standard deviations for the translation component, and thus is more likely to be the Cramér-Rao lower bound for the uncertainty of the parameters. Actually, the motion matrices were simulated using the partially exponential representation and the covariance matrices Σ_1 and Σ_2 were derived from the sample assuming the partially exponential representation and the exponential representation, respectively. This demonstrates, the meaning of the two vectors $(\boldsymbol{r}, \boldsymbol{t})$ and $(\boldsymbol{\rho}, \boldsymbol{\tau})$ are different.

Both representations are useful, as the following examples demonstrate:

1. On one hand, concatenating uncertain motions appears to be easier with the exponential representation, where the (differential of the) twist vector $\boldsymbol{s}$ of the concatenated motion $\mathsf{M} = \mathsf{M}_2\mathsf{M}_1$ is given by

$$\mathrm{d}\boldsymbol{s} = \mathrm{Ad}(\mathsf{M}_2)\,\mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2 \quad \text{with} \quad \mathrm{Ad}(\mathsf{M}_2) = \begin{bmatrix} R_2 & 0 \\ S(\boldsymbol{Z}_2)R_2 & R_2 \end{bmatrix}, \tag{18.6}$$

   or

$$\mathrm{d}\boldsymbol{r} = R_2\,\mathrm{d}\boldsymbol{r}_1 + \mathrm{d}\boldsymbol{r}_2 \quad \text{and} \quad \mathrm{d}\boldsymbol{t} = R_2\,\mathrm{d}\boldsymbol{t}_1 + \mathrm{d}\boldsymbol{t}_2 + S(\boldsymbol{Z}_2)R_2\mathrm{d}\boldsymbol{r}_2\,. \tag{18.7}$$

   Observe, the matrix $\mathrm{Ad}(\mathsf{M}_2)$ only depends on one of the two motions. In contrast, the partially exponential representation yields the joined rotation and translation components

$$\mathrm{d}\boldsymbol{\rho} = R_2\,\mathrm{d}\boldsymbol{\rho}_1 + \mathrm{d}\boldsymbol{\rho}_2 \quad \text{and} \quad \mathrm{d}\boldsymbol{\tau} = R_2\,\mathrm{d}\boldsymbol{\tau}_1 + \mathrm{d}\boldsymbol{\tau}_2 - S(R_2^\mathsf{T}\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_2\,, \tag{18.8}$$

   which looks very similar. But the relation cannot be written using a matrix only depending on one of the two motions, which is a clear disadvantage when concatenating multiple motions.

2. On the other hand the epipolar constraint for two calibrated images using partially exponential representation directly refers to the uncertain rotation and translation component

$$\mathbf{x}'^\mathsf{T}\, S(\boldsymbol{Z} + \underline{\boldsymbol{\tau}})\, R(\underline{\boldsymbol{\rho}})R\, \mathbf{x}'' = 0 \tag{18.9}$$

   whereas with the exponential representation it is given by

$$\mathbf{x}'^\mathsf{T}[I_3 \mid 0]\,\mathrm{Ad}(\underline{\mathsf{M}})^{-\mathsf{T}} \begin{bmatrix} 0 \\ I_3 \end{bmatrix} \mathbf{x}'' = 0\,, \tag{18.10}$$

   with the adjoint motion matrix $\mathrm{Ad}(\mathsf{M})$ (18.73) or more explicitly by

$$\mathbf{x}'^\mathsf{T} S(R(\underline{\boldsymbol{r}})\boldsymbol{Z} + \underline{\boldsymbol{t}})\, R(\underline{\boldsymbol{r}})R\, \mathbf{x}'' = 0 \tag{18.11}$$

   which is more cumbersome to handle, see (18.60).

When estimating a motion matrix from observed points, lines or planes using a maximum likelihood approach we basically obtain three types of numbers, which can be checked, i.e., statistically tested, using simulated data, which should lead to the following statements: (a) there are no reasons to believe the *parameters* are biased, (b) there are no reasons to believe the *variance factor*[2] deviates from 1, and (c) there are no reasons to believe the *theoretical* covariance matrix differs from the empirical covariance matrix, see (Förstner and Wrobel, 2016, Sect. 4.6.8). The test on the parameters and the covariance matrix may be performed for rotations and translations separately. Depending on how the representation for the motion is chosen and how the empirical tests are realized, the parameters usually show no bias, the variance factor does not show a deviation from 1, the covariance matrix of the rotation parameters coincide but there may be discrepancies in the covariance matrix of the translation parameters.

[2] The variance factor measures the distance of the assumed model and the given data. It is Fisher distributed, if the model holds.

The dependency on the representation of motions or poses has a direct effect on (1) checking their covariance matrices empirically, either using real or synthetic data,[3] on (2) reporting covariance matrices for motions or poses, and on (3) using them in subsequent analysis steps.

This note especially we will show:

- The Lie group property of matrix groups can, with some slight modifications be applied to the definition and use of uncertain motions represented with the partially exponential representation.
- We will discuss the variance propagation for inverse, concatenated and relative motions.
- We derive the relations for rotations as most simple case, and for motions in the mentioned two representations.
- We give two examples: (1) for estimating motions from corresponding points, and (2) deriving relative motions from bundle adjustment taking the full covariance matrix of the resulting pose parameters into account.

Basic material on Lie groups for representing uncertain transformations has been collected by Eade (2014), however, the note does not provide proofs. The most recent paper on handling uncertain motions which are correlated is by Mangelson et al. (2019), which appears not to always give the most intuitive expressions. Both papers do not address the second representation with the pair $(R, \boldsymbol{Z})$, only.

The note is organized as follows. We first give a summary of the relations, assuming the reader is acquainted with the basic concepts. Then we will provide the relations in more detail, first for rotations – as special motions –, and then for the two types of motion representations. We will compare the two motion representations and, finally, give examples for estimating motions and analysing the relative pose derived from a bundle adjustment. The proofs will be found in an appendix.

On notation. Matrices are written in capital sans-serif letters, homogeneous 4×4 matrices in upright letters, 3×3 matrices in slanted letters, such as M, A and R, S. Vectors are written in boldface times, 3-vectors representing 3D points in upright, such as $\boldsymbol{Z}$ vectors representing (numerically), numerically small entities are written in small vectors, such as $\boldsymbol{r}$ or $\boldsymbol{m}$. Stochastic entities are underscored, such as a stochastic 3×3 matrix $\underline{R}$ or a stochastic 3-vector $\underline{\boldsymbol{r}}$. Names of entities are written in calligraphic letters, in order to be able to express different representations, e.g., $\mathcal{M}(\mathsf{M})$ and $\mathcal{M}(R, \boldsymbol{Z})$. If the entity is assumed to be uncertain we underscore its name, e.g., the uncertain motion may be defined as $\underline{\mathcal{M}}(\underline{\mathsf{M}})$. For clarity, we sometimes use the multiplication dot between matrices, e.g., in the expression $\mathrm{Ad}(\mathsf{M}) \cdot \boldsymbol{s}$, which is not the multiplication dot for the scalar multiplication in $\boldsymbol{a} \cdot \boldsymbol{b}$.

18.3 Overview

We assume the following notation for Lie groups, which in our context refer to groups of regular matrices:

- A Lie group $\mathcal{G}$ has elements $g, h \in \mathcal{G}$, an operation $f = g \circ h \in \mathcal{G}$ and an inverse element such that $g^{-1} \circ g = g \circ g^{-1} = e$, with the unit element e. The dimension n of the Lie group is the number of degrees of freedom for representing an element. An element of $\mathcal{G}$ also is called an action (rotation, motion), as it is meant to operate on a vector.

[3] The author, not being aware of the difference in the two representations, spent one month finding an error in his software on estimating a motion from corresponding planes, see (Förstner and Khoshelham, 2017)

In our case we discuss rotations $\mathsf{R} \in \mathrm{SO}(3)$ and motions $\mathsf{M} \in \mathrm{SE}(3)$, having dimension $n = 3$ and $n = 6$ respectively.

- The corresponding Lie algebra $\mathcal{g}$ spans the tangent space at the unit element, its elements are n-vectors $x \in \mathbb{R}^n$ or – equivalently – matrices $X = x^\wedge$ (read: "wedge")[4] linearly depending on x, y and having the same size as the elements of the Lie group. The inverse relation is $x = X^\vee$ used for deriving the n-vector from the corresponding matrix.
  In our case the elements are 3-vectors $\boldsymbol{r} \in \mathcal{g} = \mathbb{R}^3$, also called rotation vectors, and the – not necessarily small – 6-vectors $\boldsymbol{m} \in \mathcal{g} = \mathbb{R}^6$ also called twist vectors, concatenating the rotation and the translation components of the motion. As an example for the matrix X, we have the element $\mathsf{S}(\boldsymbol{r}) = \boldsymbol{r}^\wedge$, being the skew matrix of the rotation vector $\boldsymbol{r} \in g = \mathbb{R}^3$.
- The basic relation between the Lie algebra and the Lie group is the exponential map

$$\mathcal{g} \mapsto \mathcal{G} : g = \exp(x^\wedge) \tag{18.12}$$

  which describes the elements g of $\mathcal{G}$ around the unit element e.
  As an example, we have the exponential $\mathsf{R}(\boldsymbol{r}) = \exp(\mathsf{S}(\boldsymbol{r}))$, being the rotation matrix as element of $\mathcal{G} = \mathrm{SO}(3)$. The unit element $e \in \mathcal{G}$ of the rotation group $\mathcal{G} = \mathrm{SO}(3)$ here is the unit matrix $\mathsf{I}_3 = \exp(\mathsf{S}(\mathbf{0}))$ and corresponds to the 3-vector $\mathbf{0}$, i.e., $x = 0 \in \mathcal{g}$ in the Lie algebra $\mathcal{g} = so(3)$.
  If we write $\exp(x)$, where x is an element of the Lie algebra, we actually mean $\exp(x^\wedge)$:

$$\exp(x) := \exp(x^\wedge)\,. \tag{18.13}$$

The two tables 18.1 and 18.2 collect the main algebraic relations for rotations, and motions in exponential and partially exponential representation. They are derived and discussed more in detail in the next section. The collected relations are useful in the following situations:

- Representing rotations $\mathcal{R}(\mathsf{R})$ and motions $\mathcal{M}(\mathsf{M})$ (row 1).
- Generating uncertain rotations $\underline{\mathcal{R}}(\underline{\mathsf{R}})$ and uncertain motions $\underline{\mathcal{M}}(\underline{\mathsf{M}})$ (rows 6, with 2 and 3), assuming the small elements have mean 0 and some covariance matrix. Here the difference between the exponential representation $\underline{\mathcal{M}}({}^s\underline{\mathsf{M}})$ and the partially exponential representation $\underline{\mathcal{M}}({}^\zeta\underline{\mathsf{M}})$ become visible.
- Deriving the small left rotation or motion from a small right rotation or motion leading to the same uncertain rotation or motion (rows 4 and 5), e.g., in the form $\mathsf{M}(\mathrm{Ad}(\mathsf{M})\cdot\underline{\boldsymbol{s}})\,\mathsf{M} = \mathsf{M}(\underline{\boldsymbol{s}}_{\mathrm{ad}})\cdot\mathsf{M} = \mathsf{M}\cdot\mathsf{M}(\underline{\boldsymbol{s}})$, derived from the adjoint action $\exp(x_{\mathrm{ad}}) = g\ \exp(x)g^{-1}$. Observe, the adjoint matrix is not used in other relations of the partially exponential representation.
- Deriving small deviations between estimated and true rotations and motions (row 7) using $\mathsf{V}(\boldsymbol{r}) \approx \mathsf{I}_3$ and $\mathsf{R}(\mathrm{d}\boldsymbol{r}) \approx \mathsf{I}_3 + \mathsf{S}(\mathrm{d}\boldsymbol{r})$.
- Switching between the two motion representations (row 8, columns 3 and 4).
- Deriving the mean and covariance matrix of the inverse (rows 10 and 11).
- Deriving mean and the covariance matrix of the concatenation (rows 12 and 13), of two possibly correlated rotations or motions.
- Deriving mean and the covariance matrix of the relative rotation or motion (rows 14 and 15), of two possibly correlated rotations or motions.

Comparing the relations for the two motion representations in columns 3 and 4, we observe great similarities, partially identical relations. Specifically, the two differential motions $\mathrm{d}\boldsymbol{s}$

[4] The notation results from the outer product of two vectors, which in the special case of 3-vectors reduces to the cross product. Thus we have $\boldsymbol{x} \wedge \boldsymbol{y} = \boldsymbol{x}^\wedge\,\boldsymbol{y}$ equivalent to $\boldsymbol{x} \times \boldsymbol{y} = [\boldsymbol{x}]_\times \boldsymbol{y} = \mathsf{S}(\boldsymbol{x})\boldsymbol{y}$

of the exponential representation and $\mathrm{d}\boldsymbol{\zeta}$ of the partially exponential representation are related by a linear transformation. This can be interpreted as a change of the basis of the three axes in the tangent space of the Lie group which refer to the translation component. On the other hand, is it obvious, that the relations for the exponential representation are simpler and more mutually connected. As mentioned above, e.g., the concatenation of two differential motions (row 12, column 3) only uses the adjoint matrix $\mathrm{Ad}(\mathsf{M}_2)$ of the second motion, whereas the term $-S(R_2\boldsymbol{Z}_1)$ with the skew matrix in the expression for the translation component (row 4) depends on both motions.

	1 ↓ object \ $\mathcal{G}$ →	2 SO(3)	3 SE(3), $\boldsymbol{s}$	4 SE(3), $\boldsymbol{\zeta}$
1	action, group element $g \in \mathcal{G}$	R	$\mathsf{M} = \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$	$\mathsf{M} = \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$
2	small algebra element $x \in \mathfrak{g}$	$\boldsymbol{r}$	$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{t} \end{bmatrix}$	$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\tau} \end{bmatrix}$
3	log of small action $X = x^\wedge$	$S(\boldsymbol{r})$ (18.3)	$\mathsf{A}(\boldsymbol{s}) = \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}$ (18.2)	(log of row 7, column 4)
4	adjoint action at $e \in \mathcal{G}$ $\exp(x_{\mathrm{ad}}) = g\exp(x)g^{-1}$	$R(\boldsymbol{r}_{\mathrm{ad}}) = RR(\boldsymbol{r})R^\mathsf{T}$ (18.29)	$\mathsf{M}(\boldsymbol{s}_{\mathrm{ad}}) = \mathsf{M}\,\mathsf{M}(\boldsymbol{s})\,\mathsf{M}^{-1}$ (18.70)	$\mathsf{M}(\boldsymbol{\zeta}_{\mathrm{ad}}) = \mathsf{M}\,\mathsf{M}(\boldsymbol{\zeta})\,\mathsf{M}^{-1}$ (18.91)
5	adjoint matrix for $\mathrm{d}x$ X_{ad}, $\mathrm{d}x_{\mathrm{ad}} = X_{\mathrm{ad}}\,\mathrm{d}x$	$R_{\mathrm{ad}} = R$ (18.34)	$\mathrm{Ad}(\mathsf{M}) = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})R & R \end{bmatrix}$ (18.73)	$\mathrm{Ad}(\mathsf{M}) = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})R & R \end{bmatrix}$ (18.94)
6	uncertain group element $\underline{g} = \exp(\underline{x})\,g \in \mathcal{G}$, $\underline{x}$ small	$\underline{R} = R(\underline{\boldsymbol{r}})R$ (18.21)	${}^s\underline{\mathsf{M}} = \exp(\mathsf{A}(\underline{\boldsymbol{s}}))\,\mathsf{M}$ (18.52)	${}^\zeta\underline{\mathsf{M}} = \begin{bmatrix} R(\boldsymbol{\rho})R & \boldsymbol{Z} + \underline{\boldsymbol{\tau}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$ (18.54)
7	multiplicative noise element $\exp(\underline{x}) = \underline{g}\,g^{-1} \in \mathcal{G}$	$R(\underline{\boldsymbol{r}})$ (18.56)	$\exp(\mathsf{A}(\underline{\boldsymbol{s}})) = \begin{bmatrix} R(\underline{\boldsymbol{r}}) & V(\underline{\boldsymbol{r}})\underline{\boldsymbol{t}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$ (18.56)	$\begin{bmatrix} R(\underline{\boldsymbol{\rho}}) & (I_3 - R(\underline{\boldsymbol{\rho}}))\boldsymbol{Z} + \underline{\boldsymbol{\tau}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$ (18.90)
8	differential noise element $\mathrm{d}x^{-1} \in \mathfrak{g}$	$\mathrm{d}\boldsymbol{r}$ (18.67)	$\mathrm{d}\boldsymbol{s} = \begin{bmatrix} I_3 & 0 \\ S(\boldsymbol{Z}) & I_3 \end{bmatrix} \mathrm{d}\boldsymbol{\zeta}$ (18.67)	$\mathrm{d}\boldsymbol{\zeta} = \begin{bmatrix} I_3 & 0 \\ -S(\boldsymbol{Z}) & I_3 \end{bmatrix} \mathrm{d}\boldsymbol{s}$ (18.67)

Table 18.1: Lie group elements (1/2): actions, adjoints, noisy elements, inverses, concatenations and relative actions

18.4 Uncertain Rotations

18.4.1 General setup

In all cases we represent the uncertain linear transformation $\underline{\mathcal{X}}$ by the mean transformation matrix of size $m \times m$

$$\underline{\mathcal{X}}: \quad \{X, \Sigma_{\Delta x \Delta x}\} \tag{18.14}$$

and a stochastic n-vector $\underline{\Delta\boldsymbol{x}}$, which captures the noise of the transformation, and has zero mean and a covariance matrix as second moments

$$\underline{\Delta\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\Delta x \Delta x})\,. \tag{18.15}$$

I.e. we assume the distribution is uni-modal and can be represented sufficiently well by the first two moments. We do not assume the distribution to be a normal distribution, unless we want to perform statistical testing. Then, we assume higher order of the nonlinear relations effects are small enough to be acceptable for the application.

The dimension n of the vector $\Delta\boldsymbol{x}$ is identical to the degrees of freedom of the transformation, in order to have regular covariance matrix in general. Hence, the two matrices

	1	2	3	4
	$\downarrow$ object $\setminus$ $\mathcal{G} \rightarrow$	$\mathrm{SO}(3)$	$\mathrm{SE}(3), \boldsymbol{s}$	$\mathrm{SE}(3), \boldsymbol{\zeta}$
9	uncertain inverse $\underline{g}^{(-1)} \in \mathcal{G}$	$\underline{R}^{-1} = R^{\mathsf{T}}\left(\underline{\boldsymbol{r}}^{(-1)}\right) R^{\mathsf{T}}$ (18.36)	${}^{s}\underline{\mathsf{M}}^{-1} = \mathsf{M}^{-1}\left(\underline{\boldsymbol{s}}^{(-1)}\right)\mathsf{M}^{-1}$ (18.77)	${}^{\zeta}\underline{\mathsf{M}}^{-1} = \begin{bmatrix} R(\underline{\boldsymbol{\rho}}^{(-1)})R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{Z} + \underline{\boldsymbol{\tau}}^{(-1)} \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$ (18.95)
10	differential inverse $\mathrm{d}x^{(-1)},\ \mathrm{d}x^{(-1)} \in \mathfrak{g}$	$\mathrm{d}\boldsymbol{r}^{(-1)} = -R^{\mathsf{T}}\mathrm{d}\boldsymbol{r}$ (18.38)	$\mathrm{d}\boldsymbol{s}^{(-1)} = -\mathsf{M}_{\mathrm{ad}}^{-1}\mathrm{d}\boldsymbol{s}$ (18.78)	$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}^{(-1)} \end{bmatrix} = -\begin{bmatrix} R^{\mathsf{T}} & 0 \\ R^{\mathsf{T}}S(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix}$ (18.97)
11	concatenation $g = g_2 \circ g_1 \in \mathcal{G}$	$R = R_2 R_1$	$\mathsf{M} = \mathsf{M}_2\mathsf{M}_1$	$\mathsf{M} = \mathsf{M}_2\mathsf{M}_1$
12	differential concatenation $\mathrm{d}x = \mathrm{d}(x_2 \circ x_1)$	$\mathrm{d}\boldsymbol{r} = R_{2,\mathrm{ad}}\ \mathrm{d}\boldsymbol{r}_1 + \mathrm{d}\boldsymbol{r}_2$ (18.42)	$\mathrm{d}\boldsymbol{s} = \mathrm{Ad}(\mathsf{M}_2)\ \mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2$ (18.238)	$\mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{\rho}_2 + R_2\mathrm{d}\boldsymbol{\rho}_1$ $\mathrm{d}\boldsymbol{\tau} = \mathrm{d}\boldsymbol{\tau}_2 + R_2\mathrm{d}\boldsymbol{\tau}_1 - S(R_2\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_2$ (18.102)
13	relative action $g = g_1^{-1} \circ g_2 \in \mathcal{G}$	$R = R_1^{\mathsf{T}}R_2$	$\mathsf{M} = \mathsf{M}_1^{-1}\mathsf{M}_2$	$\mathsf{M} = \mathsf{M}_1^{-1}\mathsf{M}_2$
14	differential relative action $\mathrm{d}x = \mathrm{d}(x_1^{-1} \circ x_2)$	$\mathrm{d}\boldsymbol{r} = R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{r}_2 - \mathrm{d}\boldsymbol{r}_1)$ (18.49)	$\mathrm{d}\boldsymbol{s}_{12} = \mathsf{M}_{1,\mathrm{ad}}^{-1}\ (\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1)$ (18.87)	$\mathrm{d}\boldsymbol{\rho} = R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_2 - \mathrm{d}\boldsymbol{\rho}_1)$ $\mathrm{d}\boldsymbol{\tau} = R_1^{\mathsf{T}}S(\boldsymbol{Z}_2 - \boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1 + R_1^{\mathsf{T}}\mathrm{d}(\boldsymbol{\tau}_2 - \boldsymbol{\tau}_1)$ (18.107), (18.108)

Table 18.2: Lie group elements (2/2): actions, adjoints, noisy elements, inverses, concatenations and relative actions

X and $\Sigma_{\Delta x \Delta x}$ have different dimension in general. The matrix X may be the mean motion $X := \mathbb{E}(\underline{X})$, or an estimated motion $X := \widehat{X}$, depending on the context. If we use a minimal representation, it also may be the vector $\boldsymbol{x}$ specifying the motion. The way how X is related to $\Delta\boldsymbol{x}$ needs to be specified, and even may vary for the same type of transformation. In all cases we might exploit the fact that transformations build a Lie group, i.e., a continuous group, and can be written as matrix exponential. We warm up with rotations as special motions.

18.4.2 Representing rotations

There are many ways to represent rotations. We only address three of them.

1. We start with the classical definition of rotations using Euler angles, say $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$. We generally have the uncertain rotation

$$\underline{\mathcal{R}}: \quad \{\boldsymbol{\alpha}, \Sigma_{\Delta\alpha\Delta\alpha}\}\,. \tag{18.16}$$

   e.g., specified by

$$\underline{R} = R_2(\underline{\alpha}_3)R_2(\underline{\alpha}_2)R_1(\underline{\alpha}_1) \quad \text{with} \quad \underline{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \underline{\Delta\boldsymbol{\alpha}}\,, \tag{18.17}$$

   where the indices of the rotation matrices indicate the rotation axes. In whatever sequenced the angles are applied, and what ever axis sequence is chosen, the representation for some angles will have a singularity, what is called the *gimbal lock*.

2. Therefore the Rodriguez form, depending on a rotation vector $\boldsymbol{\vartheta}$, often is preferred. Here we have the uncertain rotation

$$\underline{\mathcal{R}}: \quad \{\boldsymbol{\vartheta}, \Sigma_{\Delta\vartheta\Delta\vartheta}\}\,. \tag{18.18}$$

   It is given by the exponential map of the skew matrix S_ϑ of the rotation vector $\boldsymbol{\vartheta}$:

$$R = \exp(S(\underline{\boldsymbol{\vartheta}})) = I_3 + \frac{\sin\|\underline{\boldsymbol{\vartheta}}\|}{\|\underline{\boldsymbol{\vartheta}}\|}\underline{S}_\vartheta + \frac{1-\cos\|\underline{\boldsymbol{\vartheta}}\|}{\|\underline{\boldsymbol{\vartheta}}\|^2}\underline{S}_\vartheta^2 \quad \text{with} \quad \underline{\boldsymbol{\vartheta}} = \boldsymbol{\vartheta} + \underline{\Delta\boldsymbol{\vartheta}}\,, \tag{18.19}$$

3. Finally, we also can adopt the multiplicative definition of an uncertain rotation. Here the uncertain rotation is given by

$$\underline{\mathcal{R}}: \quad \{R, \Sigma_{rr}\}, \tag{18.20}$$

specified by

$$\underline{R} = \exp(S(\underline{\boldsymbol{r}}))\ R = R(\underline{\boldsymbol{r}})\ R\,. \tag{18.21}$$

18.4.3 Relations between the representations

When comparing the three definitions of the uncertain rotations, we need to have explicit expressions for the derivatives of R w.r.t. the elements of the noise component, either $\underline{\Delta\boldsymbol{\alpha}}$, $\underline{\Delta\boldsymbol{\theta}}$, or $\underline{\boldsymbol{r}}$.

Unfortunately, the expressions for the derivatives of the exponential $\exp(S(\boldsymbol{\vartheta}))$ w.r.t. $\boldsymbol{\vartheta}$ at some arbitrary – not necessarily small – vector, e.g., at $\boldsymbol{\vartheta} = \mathbb{E}(\underline{\boldsymbol{\vartheta}})$ are quite cumbersome. Therefore in the following we will not use the definition of an uncertain transformation using the exponential of some matrix depending on arbitrary parameters. This excludes choice 2 for defining uncertain rotations.

However, we can derive the Jacobian of the angles $\boldsymbol{r}$ in the multiplicative exponential representation w.r.t. Euler angles $\boldsymbol{\alpha}$. We specifically have

$$J_{r\alpha} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\alpha}} = [R_2(\alpha_3)R_2(\alpha_2)\boldsymbol{e}_1 \mid R_2(\alpha_3)\boldsymbol{e}_2 \mid \boldsymbol{e}_3]\ , \tag{18.22}$$

see Appendix 18.8.2. Since $|J_{r\alpha}| = \cos\alpha_2$ we have

$$J_{\alpha r} = J_{r\alpha}^{-1} \quad \text{if} \quad \cos\alpha_2 \neq 0\,. \tag{18.23}$$

This not only makes the Gimbal lock of the representation with Euler angles explicit, but shows, that we can choose either representation if we avoid the Gimbal lock.

Since all minimal representations for rotations show singularities for specific rotations or are not unique, we only discuss the option 3, with the multiplicative way to represent an uncertain rotation.

18.4.4 The rotation in exponential representation

We now discuss the adjoint rotation, the inverse, the concatenated, and the relative rotation.

18.4.4.1 The adjoint rotation

Let us for a moment define an uncertain rotation by first applying a small random rotation $R(\underline{\boldsymbol{q}})$ and then a fixed large rotation, e.g., $R := \mathbb{E}(\underline{R})$:

$$\underline{R} = R\ R(\underline{\boldsymbol{q}}) \qquad \mathbb{D}(\underline{\boldsymbol{q}}) = \Sigma_{qq}\,. \tag{18.24}$$

Applying it to a vector $\boldsymbol{x}$ we obtain a stochastic vector

$$\underline{\boldsymbol{y}} = \underline{R}\,\boldsymbol{x} = R\ R(\underline{\boldsymbol{q}})\,\boldsymbol{x}\,. \tag{18.25}$$

Now, let us choose another small rotation via what is called the *adjoint* rotation vector $\boldsymbol{q}_{\text{ad}}$ [5]

$$R(\boldsymbol{q}_{\text{ad}}) = R\ R(\boldsymbol{q})\ R^{-1} = R\ R(\boldsymbol{q})\ R^{\mathsf{T}}\,. \tag{18.26}$$

If we apply this small rotation with $\boldsymbol{q}_{\text{ad}}$ to $\boldsymbol{y} = R\boldsymbol{x}$ we obtain

$$R(\underline{\boldsymbol{q}}_{\text{ad}})\ R\,\boldsymbol{x} = R\ R(\underline{\boldsymbol{q}})\,\boldsymbol{x}\,. \tag{18.27}$$

[5] the use of the name $\boldsymbol{q}$ for a rotation vector, should not be confused with the common naming of quaternions, which do not play a role in this note.

Hence, if we first perturb $\boldsymbol{x}$ by a small rotation $\underline{\boldsymbol{q}}$ and then rotate the perturbed vector $R(\underline{\boldsymbol{q}})\,\boldsymbol{x}$ we obtain the same uncertain vector as when first rotating $\boldsymbol{x}$ and then perturbing the rotated vector $\boldsymbol{y} = R\boldsymbol{x}$ with the adjoint rotation vector $\underline{\boldsymbol{q}}_{\mathrm{ad}}$. We could also have written the relation – neutrally w.r.t. order – as

$$R(\underline{\boldsymbol{q}}_L)\; R\; \boldsymbol{x} = R\; R(\underline{\boldsymbol{q}}_R)\; \boldsymbol{x}\,, \tag{18.28}$$

the indices standing for left and right hand rotation. Hence the adjoint rotation $\underline{\boldsymbol{q}}_{\mathrm{ad}} =: \underline{\boldsymbol{q}}_L$ leads to the same result if applied to the left of a rotation as the original rotation $\boldsymbol{q}$ applied to the right of a rotation.[6]

Thus we have for any rotation vector $\boldsymbol{r}$ the adjoint rotation

$$R(\underline{\boldsymbol{r}}_{\mathrm{ad}}) = R\; R(\boldsymbol{r})\; R^{-1} \tag{18.29}$$

or the relation

$$\boxed{R(\underline{\boldsymbol{r}}_{\mathrm{ad}})\; R = R\; R(\boldsymbol{r})\,.} \tag{18.30}$$

Now, we express the *differential* adjoint rotation vector $\mathrm{d}\boldsymbol{r}_{\mathrm{ad}}$ directly as a function of the differential vector $\mathrm{d}\boldsymbol{r}$. We have

$$\mathrm{d}R(\boldsymbol{r}_{\mathrm{ad}})\; R = R\; \mathrm{d}R(\boldsymbol{r}) \tag{18.31}$$

or

$$S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}})\; R = R\; S(\mathrm{d}\boldsymbol{r}) = S(R\mathrm{d}\boldsymbol{r})\; R \tag{18.32}$$

hence

$$\boxed{\mathrm{d}\boldsymbol{r}_{\mathrm{ad}} = R\; \mathrm{d}\boldsymbol{r}\,.} \tag{18.33}$$

We observe: the differential rotation vector $\boldsymbol{r}$ and its differential adjoint rotation vector $\mathrm{d}\boldsymbol{r}_{\mathrm{ad}}$ are *linearly* related by the rotation matrix R. Since, due to $R(\boldsymbol{r}) = I_3 + S(\boldsymbol{r}) + O(r_i^2)$, the vector $\boldsymbol{r}$ spans the tangent space of a rotation at the unit rotation. But $\boldsymbol{r}_{\mathrm{ad}}$ also defines a basis, just a different one in this 3-dimensional tangent space.

Later we will see that the rotation matrix in (18.33) actually is the adjoint rotation matrix, which in this case simplifies to

$$R_{\mathrm{ad}} = R\,, \tag{18.34}$$

see (18.73).

18.4.4.2 The uncertain inverse rotation

Let now the uncertain rotation be given by

$$\boxed{\underline{R} = R(\underline{\boldsymbol{r}})\; R\,.} \tag{18.35}$$

The inverse rotation is represented the same way

$$\underline{R}^{-1} = R(\underline{\boldsymbol{r}}^{(-1)})\; R^{-1}\,. \tag{18.36}$$

The mean of the uncertain inverse is the inverse of the mean rotation:

$$\mathbb{E}(\underline{R}^{-1}) = (\mathbb{E}(\underline{R}))^{-1}\,. \tag{18.37}$$

The differential rotation vector $\mathrm{d}\boldsymbol{r}^{(-1)}$ of the inverse rotation can be shown to be

$$\boxed{\mathrm{d}\boldsymbol{r}^{(-1)} = -R^{\mathsf{T}}\mathrm{d}\boldsymbol{r}\,,} \tag{18.38}$$

see Appendix 18.8.5.

[6] Following this interpretation of the adjoint rotation it would have been straight forward to define an uncertain rotation by $\underline{R} = R\;\exp(S(\underline{\boldsymbol{r}}))$. However, most authors use the original definition of an uncertain rotation, where the noise component of the rotation is applied after the mean rotation. Unfortunately the definiton of a similarity transformation or conjugation of matrices $B = X^{-1}AX$ (https://mathworld.wolfram.com/SimilarMatrices.html) is just using the inverse operation sequence as the adjoint action $\exp(x_{\mathrm{ad}}) = g\;\exp(x)\;g^{-1}$ in a Lie group; however, see https://mathworld.wolfram.com/SimilarityTransformation.html.

18.4.4.3 The uncertain concatenated rotation

Let a *possibly correlated* rotation pair be given by

$$\left\{\underline{\mathcal{R}}_1, \underline{\mathcal{R}}_2\right\}: \quad \left\{[R_1, R_2], \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{r}}_1 \\ \underline{\boldsymbol{r}}_2 \end{bmatrix}\right)\right\}. \tag{18.39}$$

The concatenated rotation is

$$\underline{\mathcal{R}} = \underline{\mathcal{R}}_2 \underline{\mathcal{R}}_1: \quad \underline{R} = \underline{R}_2 \underline{R}_1 = R(\underline{\boldsymbol{r}}) R\,. \tag{18.40}$$

The mean of the concatenated rotations is

$$\mathbb{E}(\underline{R}) = \mathbb{E}(\underline{R}_2)\ \mathbb{E}(\underline{R}_1)\,. \tag{18.41}$$

The differential of the rotation vector $\boldsymbol{r}$ of the concatenated rotations is given by

$$\boxed{\mathrm{d}\boldsymbol{r} = R_2 \mathrm{d}\boldsymbol{r}_1 + \mathrm{d}\boldsymbol{r}_2\,.} \tag{18.42}$$

This is a special case of the concatenated motions, see Appendix 18.8.8.

Observe, Eq. (18.42) allows to derive the uncertainty of a correlated rotation pair $\left\{\underline{\mathcal{R}}_1, \underline{\mathcal{R}}_2\right\}$

$$\Sigma_{rr} = J\ \Sigma_{pp}\ J^{\mathsf{T}}\,, \tag{18.43}$$

with

$$J = [R_2 \mid I_3] \quad \text{and} \quad \Sigma_{pp} = \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{r}}_1 \\ \underline{\boldsymbol{r}}_2 \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{r_1 r_1} & \Sigma_{r_1 r_2} \\ \Sigma_{r_2 r_1} & \Sigma_{r_2 r_2} \end{bmatrix}. \tag{18.44}$$

18.4.4.4 The uncertain relative rotation

We want to determine the relative rotation

$$R_{12} = R_1^{-1} R_2 \tag{18.45}$$

in case all rotations are uncertain and possibly correlated. Let the uncertain rotations be given by

$$\underline{R}_1 = R(\underline{\boldsymbol{r}}_1)\ R_1 \quad \text{and} \quad \underline{R}_2 = R(\underline{\boldsymbol{r}}_2)\ R_2\,. \tag{18.46}$$

Then the uncertain relative rotation is

$$R(\underline{\boldsymbol{r}}_{12})\ R_{12} = (R(\underline{\boldsymbol{r}}_1)\ R_1)^{-1} R(\underline{\boldsymbol{r}}_2)\ R_2\,. \tag{18.47}$$

The mean of the relative rotations is

$$\mathbb{E}(\underline{R}) = \mathbb{E}(\underline{R}_1)^{-1}\ \mathbb{E}(\underline{R}_2)\,. \tag{18.48}$$

The differential $\mathrm{d}\boldsymbol{r}_{12}$ of the rotation vector of the relative rotation is

$$\boxed{\mathrm{d}\boldsymbol{r}_{12} = R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{r}_2 - \mathrm{d}\boldsymbol{r}_1)\,.} \tag{18.49}$$

The result is a special case of the relative motion, see Appendix 18.8.10

The result in (18.49) can be derived using the relation (18.36) for the inverse and the relation (18.42) for the concatenation.

18.5 Uncertain Motions

18.5.1 Representations

The uncertainty of a motion is captured in the uncertain twist vector $\mathit{m}(\underline{\Delta \boldsymbol{m}})$

$$\underline{\Delta \boldsymbol{m}} \sim \mathcal{N}\left(\mathbf{0}, \Sigma_{\Delta m \Delta m}\right) . \tag{18.50}$$

We address the following two representations

1. The exponential representation with the twist vector

$$\mathit{m}: \quad \boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{t} \end{bmatrix} . \tag{18.51}$$

is given by

$${}^{s}\underline{\mathcal{M}}: \quad {}^{s}\underline{\mathsf{M}} = \exp(\mathsf{A}(\underline{\boldsymbol{s}}))\ \mathsf{M} \quad \text{with} \quad \mathsf{A}(\boldsymbol{s}) = \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} . \tag{18.52}$$

2. The partially exponential representation with the twist vector

$$\mathit{m}: \quad \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\tau} \end{bmatrix} . \tag{18.53}$$

is given by

$${}^{\zeta}\underline{\mathcal{M}}: \quad {}^{\zeta}\underline{\mathsf{M}} = \begin{bmatrix} {}^{\zeta}\underline{R} & {}^{\zeta}\underline{\boldsymbol{Z}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.54}$$

with

$${}^{\zeta}R = \exp(S(\boldsymbol{\rho}))\ R \quad \text{and} \quad {}^{\zeta}\boldsymbol{Z} = \boldsymbol{Z} + \boldsymbol{\tau} \tag{18.55}$$

It appears obvious, that both representations are useful. However, they differ in the meaning of the twist vector, as we will see.

Observe, we have

$$R(\boldsymbol{r}) = \exp(S(\boldsymbol{r})) \quad \text{and} \quad \mathsf{M}(\boldsymbol{s}) = \exp(\mathsf{A}(\boldsymbol{s})) = \begin{bmatrix} R(\boldsymbol{r}) & V(\boldsymbol{r})\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.56}$$

with

$$R(\boldsymbol{r}) = I_3 + \sum_{n=1}^{\infty} \frac{S^n(\boldsymbol{r})}{n!} = I_3 + \frac{\sin\|\boldsymbol{r}\|}{\|\boldsymbol{r}\|} S_r + \frac{1-\cos\|\boldsymbol{r}\|}{\|\boldsymbol{r}\|^2} S_r^2 \tag{18.57}$$

and

$$V(\boldsymbol{r}) = I_3 + \sum_{n=1}^{\infty} \frac{S^n(\boldsymbol{r})}{(n+1)!} = I_3 + \frac{1-\cos\|\boldsymbol{r}\|}{\|\boldsymbol{r}\|^2} S_r + \frac{1-\sin\|\boldsymbol{r}\|}{\|\boldsymbol{r}\|^3} S_r^2 . \tag{18.58}$$

see Leonardos et al. (2015, eq. (19)). Thus for small values of $\|\boldsymbol{r}\|$ we may use the first order approximation

$$V(\boldsymbol{r}) \approx I_3 . \tag{18.59}$$

Therefore we have for an uncertain motion in exponential representation with small $\boldsymbol{s}$

$${}^{s}\underline{\mathsf{M}} \approx \begin{bmatrix} R(\underline{\boldsymbol{r}}) & \underline{\boldsymbol{t}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} R(\underline{\boldsymbol{r}})R & R(\underline{\boldsymbol{r}})\boldsymbol{Z} + \underline{\boldsymbol{t}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} . \tag{18.60}$$

The corresponding expression for the partially exponential representation is lengthy.

18.5.2 Comparing the two representations

We now compare the two representations with the two twist vectors

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{t} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\tau} \end{bmatrix} \tag{18.61}$$

for defining the uncertain motions as

$${}^{s}\underline{\mathsf{M}} = \exp(\mathsf{A}(\underline{\boldsymbol{s}}))\ \mathsf{M} \quad \text{and} \quad {}^{\zeta}\underline{\mathsf{M}} = \begin{bmatrix} \exp(S(\underline{\boldsymbol{\zeta}}))\ R & \boldsymbol{Z} + \underline{\boldsymbol{\tau}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{18.62}$$

Assuming the two uncertain motions are statistically equivalent, we can relate the differentials of the twist vectors. We obtain the total differential for the two motions from:

- for the exponential representation

$${}^{s}\mathrm{d}\mathsf{M} = \begin{bmatrix} S(\mathrm{d}\boldsymbol{r}) & \mathrm{d}\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} S(\mathrm{d}\boldsymbol{r})R & S(\mathrm{d}\boldsymbol{r})\boldsymbol{Z} + \mathrm{d}\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.63}$$

  and

- for the partially exponential model

$${}^{\zeta}\mathrm{d}\mathsf{M} = \begin{bmatrix} S(\mathrm{d}\boldsymbol{\rho}) & \mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} S(\mathrm{d}\boldsymbol{\rho})R & \mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.64}$$

If the uncertain motions are the same, the two differentials must be identical, and we obtain the relations

$$\mathrm{d}\boldsymbol{r} = \mathrm{d}\boldsymbol{\rho} \quad \text{or} \quad \mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{r} \tag{18.65}$$

$$\mathrm{d}\boldsymbol{t} = \mathrm{d}\boldsymbol{\tau} + S(\boldsymbol{Z})\mathrm{d}\boldsymbol{\rho} \quad \text{or} \quad \mathrm{d}\boldsymbol{\tau} = \mathrm{d}\boldsymbol{t} - S(\boldsymbol{Z})\mathrm{d}\boldsymbol{r} \tag{18.66}$$

$$\begin{bmatrix} \mathrm{d}\boldsymbol{r} \\ \mathrm{d}\boldsymbol{t} \end{bmatrix} = \begin{bmatrix} I_3 & 0 \\ S(\boldsymbol{Z}) & I_3 \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} I_3 & 0 \\ -S(\boldsymbol{Z}) & I_3 \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{r} \\ \mathrm{d}\boldsymbol{t} \end{bmatrix}$$

between the twist vectors $\boldsymbol{s}$ and $\boldsymbol{\zeta}$. Hence, we have the relations

$$\boxed{\mathrm{d}\boldsymbol{s} = J_{s\zeta}\mathrm{d}\boldsymbol{\zeta} \quad \text{and} \quad \mathrm{d}\boldsymbol{\zeta} = J_{\zeta s}\mathrm{d}\boldsymbol{s}} \tag{18.67}$$

with

$$J_{s\zeta} = \begin{bmatrix} I_3 & 0 \\ S(\boldsymbol{Z}) & I_3 \end{bmatrix} \quad \text{and} \quad J_{\zeta s} = J_{s\zeta}^{-1} = \begin{bmatrix} I_3 & 0 \\ -S(\boldsymbol{Z}) & I_3 \end{bmatrix}. \tag{18.68}$$

This allows us to transfer the covariance matrices of the twist vectors

$$\Sigma_{s\xi} = J_{s\zeta}\Sigma_{\zeta\zeta}J_{s\zeta}^{\mathsf{T}} \quad \text{and} \quad \Sigma_{\zeta\zeta} = J_{s\zeta}\Sigma_{ss}J_{s\zeta}^{\mathsf{T}}. \tag{18.69}$$

between both representations.

As a result, we find: the uncertain rotation components $\mathrm{d}\boldsymbol{r}$ and $\mathrm{d}\boldsymbol{\rho}$ of both representations are identical but the uncertain translation components $\mathrm{d}\boldsymbol{t}$ and $\mathrm{d}\boldsymbol{\tau}$ differ by the effect of the uncertain rotation applied to the full translation $\boldsymbol{Z}$.

18.5.3 The motion in exponential representation

We now discuss the adjoint, the inverse, the concatenated, and the relative motion in exponential representation.

18.5.3.1 The adjoint motion for the exponential representation

The adjoint motion $\mathsf{M}(\boldsymbol{s}_{\mathrm{ad}})$ is defined with, what is called the *adjoint* motion vector $\boldsymbol{s}_{\mathrm{ad}}$,

$$\mathsf{M}(\boldsymbol{s}_{\mathrm{ad}}) = \mathsf{M}\,\mathsf{M}(\boldsymbol{s})\,\mathsf{M}^{-1}\,. \tag{18.70}$$

For proofs we often use it in the differential form

$$\mathsf{A}(\boldsymbol{s}_{\mathrm{ad}})\,\mathsf{M} = \mathsf{M}\,\mathsf{A}(\boldsymbol{s})\,, \tag{18.71}$$

allowing to exchange the differential of the perturbing noise matrix A and the motion matrix M. Also here we obtain a simple linear relation between the differentials of the twist vectors

$$\boxed{\mathrm{d}\boldsymbol{s}_{\mathrm{ad}} = \mathrm{Ad}(\mathsf{M})\,\mathrm{d}\boldsymbol{s}\,,} \tag{18.72}$$

with the adjoint motion matrix relating the two 6-vectors

$$\boxed{\mathrm{Ad}(\mathsf{M}) = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})R & R \end{bmatrix},} \tag{18.73}$$

and its inverse

$$\mathsf{M}_{\mathrm{ad}}^{-1} = \begin{bmatrix} R^{\mathsf{T}} & 0 \\ R^{\mathsf{T}}\,S^{\mathsf{T}}(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix}. \tag{18.74}$$

Eq. (18.73) can also be written as

$$\boxed{\mathsf{A}\left(\mathrm{Ad}(\mathsf{M})\cdot\mathrm{d}\boldsymbol{s}\right)\ \ \mathsf{M} = \mathsf{M}\,\mathsf{A}(\mathrm{d}\boldsymbol{s})\,.} \tag{18.75}$$

The proof is given in Appendix 18.8.3. We observe: the differential rotation vector $\boldsymbol{s}$ and its differential adjoint motion vector $\mathrm{d}\boldsymbol{s}_{\mathrm{ad}}$ are linearly related by the adjoint motion matrix $\mathrm{Ad}(\mathsf{M})$. The relation between the small motion vectors only holds for differential motions. This is sufficient for all practical cases, where the relative precision of the motion parameters is high enough. Observe, when restricting to rotations we have

$$R_{\mathrm{ad}} = R\,, \tag{18.76}$$

The simplicity of this relation does not reveil the strength of the concept for more general transformations.

18.5.3.2 The inverse motion in exponential representation

Similarly as for rotations, we can derive the relation between the differential twist vector of the inverse motion to the one of the original motion.

We have the basic relation

$$\underline{\mathsf{M}}^{-1} = \mathsf{M}\left(\underline{\boldsymbol{s}}^{(-1)}\right)\cdot\mathsf{M}^{-1} = \left(\mathsf{M}(\underline{\boldsymbol{s}})\cdot\mathsf{M}\right)^{-1} = \mathsf{M}^{-1}\cdot\mathsf{M}^{-1}(\underline{\boldsymbol{s}})\,. \tag{18.77}$$

Using the adjoint motion we can derive the following relation between the differential twist vectors:

$$\boxed{\mathrm{d}\boldsymbol{s}^{(-1)} = -\mathrm{Ad}(\mathsf{M})^{-1}\,\mathrm{d}\boldsymbol{s}\,,} \tag{18.78}$$

see the proof in the Appendix 18.8.6

18.5.3.3 The concatenated motion in exponential representation

Let a possibly correlated motion pair be given by

$$\left\{{}^{s}\underline{\mathcal{M}}_1, {}^{s}\underline{\mathcal{M}}_2\right\}: \quad \left\{[\mathsf{M}_1, \mathsf{M}_2], \mathbb{D}\left(\begin{bmatrix} \underline{\boldsymbol{s}}_1 \\ \underline{\boldsymbol{s}}_2 \end{bmatrix}\right)\right\}. \tag{18.79}$$

The concatenated motion is

$$ {}^{s}\underline{\mathcal{M}} = {}^{s}\underline{\mathcal{M}}_2 \, {}^{s}\underline{\mathcal{M}}_1 : \quad {}^{s}\underline{\mathsf{M}} = {}^{s}\underline{\mathsf{M}}_2 {}^{s}\underline{\mathsf{M}}_1 = \mathsf{M}(\underline{\boldsymbol{s}})\mathsf{M} \,. \tag{18.80} $$

The mean of the concatenated motion is

$$ \mathbb{E}({}^{s}\underline{\mathsf{M}}) = \mathbb{E}({}^{s}\underline{\mathsf{M}}_2) \, \mathbb{E}({}^{s}\underline{\mathsf{M}}_1) \,. \tag{18.81} $$

The differential of the rotation vector $\boldsymbol{s}$ of the concatenated rotation is given by

$$ \boxed{\mathrm{d}\boldsymbol{s} = \mathsf{M}_2 \mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2 \,.} \tag{18.82} $$

see Appendix 18.8.8.

18.5.3.4 The relative motion in exponential representation

We want to determine the relative motion

$$ \mathsf{M}_{12} = \mathsf{M}_1^{-1} \mathsf{M}_2 \tag{18.83} $$

in case all motions are uncertain. Let the uncertain motions be given by

$$ {}^{s}\underline{\mathsf{M}}_1 = \mathsf{M}(\underline{\boldsymbol{s}}_1) \, \mathsf{M}_1 \quad \text{and} \quad {}^{s}\underline{\mathsf{M}}_2 = \mathsf{M}(\underline{\boldsymbol{s}}_2) \, \mathsf{M}_2 \,. \tag{18.84} $$

Then the uncertain relative motion is

$$ \mathsf{M}(\underline{\boldsymbol{s}}_{12}) \, \mathsf{M}_{12} = (\mathsf{M}(\underline{\boldsymbol{s}}_1) \, \mathsf{M}_1)^{-1} \mathsf{M}(\underline{\boldsymbol{s}}_2) \, \mathsf{M}_2 \,., \tag{18.85} $$

The mean of the relative rotations is

$$ \mathbb{E}({}^{s}\underline{\mathsf{M}}) = \mathbb{E}({}^{s}\underline{\mathsf{M}}_1)^{-1} \, \mathbb{E}({}^{s}\underline{\mathsf{M}}_2) \,. \tag{18.86} $$

The differential $\mathrm{d}\boldsymbol{s}_{12}$ of the rotation vector of the relative rotation is

$$ \boxed{\mathrm{d}\boldsymbol{s}_{12} = \mathsf{M}_1^{\mathsf{T}} (\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1) \,.} \tag{18.87} $$

see Appendix 18.8.10

18.5.4 The motion in partially exponential representation

The uncertain motion is defined as

$$ {}^{\zeta}\underline{\mathsf{M}} = \begin{bmatrix} R(\boldsymbol{\rho})R & \boldsymbol{Z} + \underline{\boldsymbol{\tau}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} , \qquad \mathbb{D}(\underline{\boldsymbol{\zeta}}) = \Sigma_{\zeta\zeta} \,, \tag{18.88} $$

We also can write this as a multiplication of a motion with a small random motion

$$ {}^{\zeta}\underline{\mathsf{M}} = \mathsf{M}(\underline{\boldsymbol{\zeta}}) \, \mathsf{M} \quad \text{with} \tag{18.89} $$

with the small motion

$$ \mathsf{M}(\boldsymbol{\zeta}) = \begin{bmatrix} R(\boldsymbol{\rho}) & (I_3 - R(\boldsymbol{\rho}))\boldsymbol{Z} + \boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \quad \text{and} \quad \mathsf{M} = \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.90} $$

18.5.4.1 The adjoint motion for the partially exponential representation

Since the adjoint motion transfers small motions, we also can define an adjoint motion in case of the partially exponential representation. It is defined as the motion depending on the adjoint twist vector $\boldsymbol{\zeta}_{\mathrm{ad}}$

$$\mathsf{M}(\boldsymbol{\zeta}_{\mathrm{ad}}) = \mathsf{M}\,\mathsf{M}(\boldsymbol{\zeta})\,\mathsf{M}^{-1}\,. \tag{18.91}$$

Thus we have the form which can be used in proofs

$$\boxed{\mathsf{M}(\underline{\boldsymbol{\zeta}}_{\mathrm{ad}})\,\mathsf{M} = \mathsf{M}\,\mathsf{M}(\boldsymbol{\zeta})\,.} \tag{18.92}$$

Interestingly, also here we have a linear relationship between the differential adjoint twistvector $\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}}$ and the differential original twist vector $\mathrm{d}\boldsymbol{\zeta}$:

$$\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}} = {}^{\zeta}\mathsf{M}_{\mathrm{ad}}\,\mathrm{d}\boldsymbol{\zeta} \tag{18.93}$$

with the adjoint motion matrix

$$\boxed{{}^{\zeta}\mathsf{M}_{\mathrm{ad}} = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})\,R & R \end{bmatrix},} \tag{18.94}$$

Observe, that the two adjoint matrices ${}^{s}\mathsf{M}_{\mathrm{ad}}$ in (18.73) and ${}^{\zeta}\mathsf{M}_{\mathrm{ad}}$ in (18.94) are identical. This results from the fact, that the adjoint motion for a differential twist has translation component zero, hence the two adjoint twist vectors do not differ if the original twist vectors are the same: The Jacobians in (18.68) then are unit matrices. This is the reason, why we did not indicate the difference in the naming of the adjoint matrices in Table 18.1 in row 5, columns 3 and 4.

18.5.4.2 The inverse motion in partially exponential representation

The uncertain inverse in partially exponential representation is defined as

$${}^{\zeta}\underline{\mathsf{M}}^{-1} = \begin{bmatrix} R(\underline{\boldsymbol{\rho}}^{(-1)})R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{Z} + \underline{\boldsymbol{\tau}}^{(-1)} \\ 0 & 1 \end{bmatrix}, \tag{18.95}$$

and depends on the stochastic twist vector

$$\underline{\boldsymbol{\zeta}}^{(-1)} = \begin{bmatrix} \underline{\boldsymbol{\rho}}^{(-1)} \\ \underline{\boldsymbol{\tau}}^{(-1)} \end{bmatrix}. \tag{18.96}$$

As we saw in the last section, the differential adjunct twists are related to their twists via the adjoint motion matrix, which is identical for both cases. Therefore also the differential of the inverse twist vector in the partially exponential representation is given by

$$\boxed{\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}^{(-1)} \end{bmatrix} = -\begin{bmatrix} R^{\mathsf{T}} & 0 \\ -R^{\mathsf{T}}S^{\mathsf{T}}(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix},} \tag{18.97}$$

see Appendix 18.8.7. Observe, this is not the negative inverse of the adjunct motion matrix, since we have

$$\mathrm{Ad}(\mathsf{M})\,\mathrm{Ad}(\mathsf{M})^{-1} = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})\,R & R \end{bmatrix}\begin{bmatrix} R^{\mathsf{T}} & 0 \\ -R^{\mathsf{T}}S(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix} = \begin{bmatrix} I_3 & 0 \\ 0 & I_3 \end{bmatrix}, \tag{18.98}$$

and the second factor differs in the sign of the (2,1)-submatrix.

18.5.4.3 The concatenated motion in partially exponential representation

Let a possibly correlated motion pair be given by

$$\{{}^{\zeta}\underline{\mathcal{M}}_1, {}^{\zeta}\underline{\mathcal{M}}_2\} : \quad \left\{[\mathsf{M}_1, \mathsf{M}_2], \mathbb{D}\left(\begin{bmatrix} \underline{\zeta}_1 \\ \underline{\zeta}_2 \end{bmatrix}\right)\right\} . \tag{18.99}$$

The concatenated motion is

$${}^{\zeta}\underline{\mathcal{M}} = {}^{\zeta}\underline{\mathcal{M}}_2 {}^{\zeta}\underline{\mathcal{M}}_1 : \quad {}^{\zeta}\underline{\mathsf{M}} = {}^{\zeta}\underline{\mathsf{M}}_2 \, {}^{\zeta}\underline{\mathsf{M}}_1 . \tag{18.100}$$

We find the mean values of the concatenated motion is

$$\mathbb{E}({}^{\zeta}\underline{\mathsf{M}}) = \mathbb{E}\left({}^{\zeta}\underline{\mathsf{M}}_2\right) \; \mathbb{E}\left({}^{\zeta}\underline{\mathsf{M}}_1\right) . \tag{18.101}$$

The differentials of the twist vectors also are linearly related by

$$\boxed{\mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{\rho}_2 + R_2 \mathrm{d}\boldsymbol{\rho}_1 \quad \text{and} \quad \mathrm{d}\boldsymbol{\tau} = \mathrm{d}\boldsymbol{\tau}_2 + R_2 \mathrm{d}\boldsymbol{\tau}_1 - S(R_2 \boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_2} \tag{18.102}$$

Observe, the rotation component transforms as for the exponential represenation and the translation component has a different term with the skew matrix. Moreover, and much more important: this matrix depends on both motions via $\boldsymbol{Z}_1$ and R_2, which complicates multiple concatenations.

18.5.4.4 The relative motion in partially exponential representation

Let a possibly correlated motion pair be given by

$$\{{}^{\zeta}\underline{\mathcal{M}}_1, {}^{\zeta}\underline{\mathcal{M}}_2\} : \quad \left\{[\mathsf{M}_1, \mathsf{M}_2], \mathbb{D}\left(\begin{bmatrix} \underline{\zeta}_1 \\ \underline{\zeta}_2 \end{bmatrix}\right)\right\} . \tag{18.103}$$

Then the relative pose can be determined by

$${}^{\zeta}\underline{\mathsf{M}}_{12} = {}^{\zeta}\underline{\mathsf{M}}_1^{-1} {}^{\zeta}\underline{\mathsf{M}}_2 = \begin{bmatrix} {}^{\zeta}\underline{R}_{12} & {}^{\zeta}\underline{\boldsymbol{Z}}_{12} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_{12})R_{12} & \boldsymbol{Z}_{12} + \underline{\boldsymbol{\tau}}_{12} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} . \tag{18.104}$$

or from

$${}^{\zeta}\underline{R}_{12} = {}^{\zeta}\underline{R}_1^{\mathsf{T}} {}^{\zeta}\underline{R}_2 \quad \text{and} \quad {}^{\zeta}\underline{\boldsymbol{Z}}_{12} = {}^{\zeta}\underline{R}_1^{\mathsf{T}}({}^{\zeta}\underline{\boldsymbol{Z}}_2 - {}^{\zeta}\underline{\boldsymbol{Z}}_1) . \tag{18.105}$$

We obtain the mean relative motion as

$$\mathbb{E}({}^{\zeta}\underline{\mathsf{M}}_{12}) = \mathbb{E}({}^{\zeta}\underline{\mathsf{M}}_1)^{-1}\mathbb{E}({}^{\zeta}\underline{\mathsf{M}}_2) \tag{18.106}$$

Using the result from the uncertain relative rotation the differentials of the rotation and the translation vector are related by

$$\boxed{\mathrm{d}\boldsymbol{\rho}_{12} = R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_2 - \mathrm{d}\boldsymbol{\rho}_1)} \tag{18.107}$$

and by variance propagation from (18.105)

$$\boxed{\mathrm{d}\boldsymbol{\tau}_{12} = R_1^{\mathsf{T}} S(\boldsymbol{Z}_2 - \boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1 + R_1^{\mathsf{T}}\mathrm{d}(\boldsymbol{\tau}_2 - \boldsymbol{\tau}_1) .} \tag{18.108}$$

18.5.5 Evaluating the covariance matrix of estimated motions

We now discuss how to evaluate whether a theoretical covariance matrix is consistent with an empirical one.

Evaluating whether the theoretical covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$ of estimated parameters $\boldsymbol{\theta}$ is trustworthy, it can be compared with the empirical covariance matrix $\widehat{\Sigma}_{\hat{\theta}\hat{\theta}}$ derived from

a sample of $\{\hat{\boldsymbol{\theta}}_k, k = 1, ..., K\}$, when knowing the true value $\tilde{\boldsymbol{\theta}}$, e.g., when using simulated data

$$\hat{\Sigma}_{\hat{\theta}\hat{\theta}} = \frac{1}{K} \sum_{k=1}^{K} (\hat{\boldsymbol{\theta}}_k - \tilde{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}}_k - \tilde{\boldsymbol{\theta}})^\mathsf{T} . \tag{18.109}$$

In our context we, instead of the differences $\hat{\boldsymbol{\theta}}_k - \tilde{\boldsymbol{\theta}}$ of the estimated and the true parameters we use the estimated twist vectors $\widehat{\boldsymbol{m}}_k$, since their means are zero.

When evaluating the covariance matrix of estimated motions from a sample $\hat{\mathsf{M}}_k, k = 1, ..., K$ and a given true motion M we need to distinguish how we determine the empirical covariance matrix of the twist vector.

- In the case of the exponential representation we use the small matrices

$$\begin{aligned}
\mathsf{L}_k &= \hat{\mathsf{M}}_k \mathsf{M}^{-1} && (18.110)\\
&= \begin{bmatrix} \hat{R}_k & \hat{\boldsymbol{Z}}_k \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} R^\mathsf{T} & -R^\mathsf{T}\boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (18.111)\\
&= \begin{bmatrix} \hat{R}_k R^\mathsf{T} & \hat{\boldsymbol{Z}}_k - \hat{R}_k R^\mathsf{T} \boldsymbol{Z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (18.112)\\
&\approx I_4 + \begin{bmatrix} S(\hat{\boldsymbol{r}}_k) & \hat{\boldsymbol{t}}_k \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . && (18.113)
\end{aligned}$$

  and derive the small twist vectors $\hat{\boldsymbol{s}}_k = (\hat{\boldsymbol{r}}_k, \hat{\boldsymbol{t}}_k)$ from

$$\hat{\boldsymbol{r}}_k = \begin{bmatrix} L_{k23} \\ L_{k12} \\ L_{k31} \end{bmatrix} = \begin{bmatrix} (\hat{R}_k R^\mathsf{T})_{23} \\ (\hat{R}_k R^\mathsf{T})_{12} \\ (\hat{R}_k R^\mathsf{T})_{31} \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{t}}_k = \begin{bmatrix} L_{k14} \\ L_{k24} \\ L_{k34} \end{bmatrix} = \hat{\boldsymbol{Z}}_k - \hat{R}_k R^\mathsf{T} \boldsymbol{Z} \tag{18.114}$$

  This also could be written compactly as

$$\hat{\boldsymbol{s}}_k = \log \left(\hat{\mathsf{M}}_k \mathsf{M}^{-1} \right)^\vee , \tag{18.115}$$

  the operator $^\vee$ (read: "vee") being the inverse of the operator $^\wedge$, thus, if $X = x^\wedge$ we have $x = X^\vee$.
  Then the empirical covariance matrix of $\underline{\hat{\boldsymbol{s}}}$ is

$$\hat{\Sigma}_{\hat{s}\hat{s}} = \frac{1}{K} \sum_k \boldsymbol{s}_k \boldsymbol{s}_k^\mathsf{T} . \tag{18.116}$$

- In the case of the partially multiplicative model we use

$$G_k = \hat{R}_k R^\mathsf{T} \approx I_3 + S(\boldsymbol{\rho}_k) \quad \text{and} \quad \boldsymbol{h}_k = \hat{\boldsymbol{Z}}_k - \boldsymbol{Z} = \boldsymbol{\tau}_k \tag{18.117}$$

  This leads to the elements of the small twist vector $\boldsymbol{\zeta}_k = (\boldsymbol{\rho}_k, \boldsymbol{\tau}_k)$

$$\boldsymbol{\rho}_k = \begin{bmatrix} G_{k23} \\ G_{k12} \\ G_{k31} \end{bmatrix} = \begin{bmatrix} (\hat{R}_k R^\mathsf{T})_{23} \\ (\hat{R}_k R^\mathsf{T})_{12} \\ (\hat{R}_k R^\mathsf{T})_{31} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\tau}_k = \boldsymbol{h}_k = \hat{\boldsymbol{Z}}_k - \boldsymbol{Z} . \tag{18.118}$$

  Then the empirical covariance matrix of $\underline{\hat{\boldsymbol{s}}}$ is

$$\hat{\Sigma}_{\hat{\zeta}\hat{\zeta}} = \frac{1}{I} \sum_k \boldsymbol{\zeta}_k \boldsymbol{\zeta}_k^\mathsf{T} . \tag{18.119}$$

As a result, linearizing the given model and deriving the empirical deviations of the estimated motions from the true motion need to be consistent.

In both cases we use a statistical test to check whether the expectation of the covariance matrix from the sample is identical to the theoretical covariance matrix, see (Förstner and Wrobel, 2016, Sect. 4.6.8.2).

18.6 Examples

We discuss two applications:

- Estimating motion parameters,
- Comparing absolute and relative poses.

18.6.1 Estimating motion parameters

Let us assume we have given I corresponding 3D points $\{\boldsymbol{X}, \boldsymbol{Y}\}_i, i = 1, ..., I$, where the coordinates $\boldsymbol{X}_i$ are fixed given values, and the coordinates $\boldsymbol{Y}_i$ are noisy observations of the corresponding moved points $\boldsymbol{X}_i$, having covariance matrices Σ_{ii}. We assume the correspondences are mutually independent, hence $\Sigma_{ii'} = \boldsymbol{0}$. Then, with the homogeneous coordinates

$$\mathbf{X}_i = \begin{bmatrix} \boldsymbol{X}_i \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_i = \begin{bmatrix} \boldsymbol{Y}_i \\ 1 \end{bmatrix} \tag{18.120}$$

we have the non-linear Gauss-Markov model (stochastical variables are underscored)

$$\mathbb{E}(\underline{\mathbf{Y}}_i) = \mathsf{M}\ \mathbf{X}_i \quad \text{and} \quad \mathbb{D}(\underline{\mathbf{Y}}_i) = \begin{bmatrix} \Sigma_{ii} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \quad \text{with} \quad i = 1, ..., I\,. \tag{18.121}$$

or, with the residuals (corrections),

$$\mathbf{Y}_i + \mathbf{v}_i = \mathsf{M}\ \mathbf{X}_i\,. \tag{18.122}$$

We assume we have an approximate motion matrix M^a. The model needs to be linearized, which depends on the type of representation.

18.6.1.1 Linearization with the exponential representation

With the exponential representation we have

$$\mathbf{Y}_i + \mathbf{v}_i = \mathsf{M}(\boldsymbol{s})\ \mathsf{M}^a\ \mathbf{X}_i = \mathsf{M}(\boldsymbol{s})\ \mathbf{X}_i^a \tag{18.123}$$

with the approximately *moved* coordinates

$${}^s\mathbf{X}_i^a = \mathsf{M}^a\ \mathbf{X}_i\,. \tag{18.124}$$

The goal is to estimate the twist vector $\boldsymbol{s}$ form the I correspondences. Linearization leads to

$$\mathbf{Y}_i + \mathbf{v}_i = (I_4 + A(\boldsymbol{s}))\ {}^s\mathbf{X}_i^a \tag{18.125}$$

where $\mathbf{v}_i$ are the residuals of (corrections to) the coordinates $\mathbf{Y}_i$. With the linearized observations

$${}^s\Delta\mathbf{y} = \mathbf{Y}_i - {}^s\mathbf{X}_i^a \tag{18.126}$$

this can be rewritten as

$$\begin{aligned}
{}^s\Delta\mathbf{y}_i + \mathbf{v}_i &= A(\boldsymbol{s}))\ {}^s\mathbf{X}_i^a && (18.127)\\
{}^s\Delta\mathbf{y}_i + \mathbf{v}_i &= \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} {}^s\boldsymbol{X}_i^a \\ 1 \end{bmatrix} && (18.128)\\
{}^s\Delta\boldsymbol{y}_i + \boldsymbol{v}_i &= S(\boldsymbol{r})\,{}^s\boldsymbol{X}_i^a + \boldsymbol{t} && (18.129)\\
& && (18.130)
\end{aligned}$$

thus finally

$${}^s\Delta\boldsymbol{y}_i + \mathbf{v}_i = [-S({}^sX_i^a) \mid I_3] \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{t} \end{bmatrix} \tag{18.131}$$

or the linear substitute model

$$\Delta \boldsymbol{y}_i + \mathbf{v}_i = {}^{s}\mathsf{X}_i \, \Delta\boldsymbol{\theta} \tag{18.132}$$

with the design matrix for each point and the unknown parameters

$${}^{s}\mathsf{X}_i = [-\mathsf{S}({}^{s}\boldsymbol{X}_i^a) \mid \mathsf{I}_3] \quad \text{and} \quad \Delta\boldsymbol{\theta} = \boldsymbol{s}\,. \tag{18.133}$$

The update of the parameters within the ν-th iteration is

$${}^{s}\mathsf{M}^{(\nu+1)} = \begin{bmatrix} \exp(\mathsf{S}(\widehat{\boldsymbol{r}}^{(\nu)})) & \widehat{\boldsymbol{t}}^{(\nu)} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathsf{M}^{(\nu)}\,. \tag{18.134}$$

18.6.1.2 Linearization with the partially exponential representation

With the exponential representation we have

$$\mathbf{Y}_i + \mathbf{v}_i = {}^{\zeta}\mathsf{M}\ \mathbf{X}_i = \begin{bmatrix} \mathsf{R}(\boldsymbol{\rho})\ \mathsf{R} & \boldsymbol{Z} + \boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i \\ 1 \end{bmatrix} \tag{18.135}$$

Linerization leads to

$$\mathbf{Y}_i + \mathbf{v}_i = \begin{bmatrix} (\mathsf{I}_3 + \mathsf{S}(\boldsymbol{\rho}))\ \mathsf{R}^a & \boldsymbol{Z}^a + \boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i \\ 1 \end{bmatrix} \tag{18.136}$$

$$\mathbf{Y}_i + \mathbf{v}_i = \begin{bmatrix} \mathsf{R}^a\ \boldsymbol{X}_i + \mathsf{S}(\boldsymbol{\rho})\ \mathsf{R}^a\ \boldsymbol{X}_i + \boldsymbol{Z}^a + \boldsymbol{\tau} \\ 1 \end{bmatrix} \tag{18.137}$$

With the approximately *rotated* coordinates

$${}^{\zeta}\boldsymbol{X}_i^a = \mathsf{R}^a\ \boldsymbol{X}_i \tag{18.138}$$

and the linearized observations

$${}^{\zeta}\Delta\boldsymbol{y} = \boldsymbol{Y}_i - (\mathsf{R}^a\ \boldsymbol{X}_i + \boldsymbol{Z}^a) \tag{18.139}$$

we have the linearized model

$${}^{\zeta}\Delta\boldsymbol{y} + {}^{\zeta}\boldsymbol{v}_i = \mathsf{S}(\boldsymbol{\rho}){}^{\zeta}\boldsymbol{X}_i^a + \boldsymbol{\tau} \tag{18.140}$$

or finally

$${}^{\zeta}\Delta\boldsymbol{y} + {}^{\zeta}\boldsymbol{v}_i = {}^{\zeta}\mathsf{X}_i \, \Delta\boldsymbol{\theta} \tag{18.141}$$

with

$${}^{\zeta}\mathsf{X}_i = [-\mathsf{S}({}^{\zeta}\boldsymbol{X}_i^a) \mid \mathsf{I}_3] \quad \text{and} \quad \Delta\boldsymbol{\theta} = \boldsymbol{\zeta}\,. \tag{18.142}$$

The update of the parameters within the ν-th iteration is

$${}^{\zeta}\mathsf{M}^{(\nu+1)} = \begin{bmatrix} \mathsf{R}(\widehat{\boldsymbol{\rho}}^{(\nu)})\ \mathsf{R}^{(\nu)} & \boldsymbol{Z}^{(\nu)} + \widehat{\boldsymbol{\tau}}^{(\nu)} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{18.143}$$

18.6.1.3 Comparison

The design matrices differ in the argument of the skew matrix. For the exponential model we have explicitly

$${}^{s}\mathsf{X}_i = [-\mathsf{S}(\mathsf{R}^a\boldsymbol{X}_i + \boldsymbol{Z}^a) \mid \mathsf{I}_3] \tag{18.144}$$

while for the partially exponential model we have

$${}^{\zeta}\mathsf{X}_i = [-\mathsf{S}(\mathsf{R}^a\boldsymbol{X}_i) \mid \mathsf{I}_3] \tag{18.145}$$

Hence, the normal equation matrices

$${}^{s}\mathsf{N} = \sum_i {}^{s}\mathsf{X}_i^{\mathsf{T}}\ \Sigma_{ii}^{-1}\ {}^{s}\mathsf{X}_i \quad \text{and} \quad {}^{\zeta}\mathsf{N} = \sum_i {}^{\zeta}\mathsf{X}_i^{\mathsf{T}}\ \Sigma_{ii}^{-1}\ {}^{\zeta}\mathsf{X}_i \tag{18.146}$$

are differing in the rotation component and therefore also the inverse normal equation matrices, i.e., the covariance matrices of the estimated parameters.

Observe, in an extended Kalman filter for the motion parametrized by $\boldsymbol{x}$ with innovation of measurement residual $\boldsymbol{y}_k = \boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})$ the Jacobian $H = \partial \boldsymbol{h}/\partial \boldsymbol{x}$ depends on the representation of the motion in the function $\boldsymbol{h}$, which may use one of the representations discussed in this note. The resulting covariance matrices will of course differ, depending on the choice of the representation.

18.6.2 Example for comparing absolute and relative poses in multi-view analysis

Let aus assume a free bundle block adjustment with two cameras at $\boldsymbol{Z}_t, t = 1, 2$ and 6 scene points $\boldsymbol{X}_i, i = 1..6$, as shown in Fig 18.1. The basis points towards the scene points, mimicking a docking situation. We are interested in precision of the relative motion.

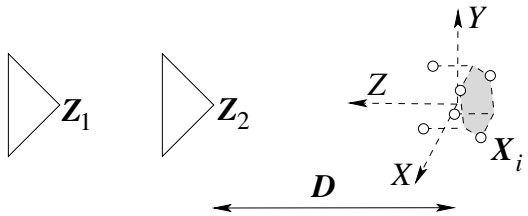


Figure 18.1: Relative motion from free bundle adjustment. The basis is 1 m. The distance D to the scene points is 2 m. The distance difference of the scene points is 0.3 m. The uncertainty of the image rays is 0.1 mrad

The free bundle adjustment with the software package BACS[7] (Schneider and Förstner, 2013) yields the covariance matrix of all pose parameters fixing the gauge in the centroid of the given scene points. The covariance matrix of the 12 parameters of the two twists is given by

$$\Sigma_{\hat{p}\hat{p}} = \mathbb{D}\left(\begin{bmatrix} \underline{\zeta}_1 \\ \underline{\zeta}_2 \end{bmatrix}\right) = SRS \tag{18.147}$$

where the diagonal matrix $S = \mathrm{Diag}([\sigma_{p_u}])$ contains the standard deviations, and the matrix $R = [\rho_{u'u''}]$ the correlations between the parameters. As an example we obtain the standard deviations for the rotations in [rad] and for the translations in [m]

$$\begin{bmatrix} \sigma_{\rho 11} & \sigma_{\tau 11} & \sigma_{\rho 21} & \sigma_{\tau 21} \\ \sigma_{\rho 12} & \sigma_{\tau 12} & \sigma_{\rho 22} & \sigma_{\tau 22} \\ \sigma_{\rho 13} & \sigma_{\tau 13} & \sigma_{\rho 23} & \sigma_{\tau 23} \end{bmatrix} = \begin{bmatrix} 0.0141 & 0.0425 & 0.0137 & 0.0278 \\ 0.0141 & 0.0425 & 0.0137 & 0.0278 \\ 0.0004 & 0.0121 & 0.0003 & 0.0078 \end{bmatrix} \tag{18.148}$$

The correlation matrix R is given by

$$\frac{1}{1000}\begin{bmatrix}
1000 & 0 & 0 & 0 & -1000 & 0 & 995 & 0 & 0 & 0 & -995 & 0 \\
0 & 1000 & 0 & 1000 & 0 & 0 & 0 & 995 & 0 & 995 & 0 & 0 \\
0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\
0 & 1000 & 0 & 1000 & 0 & 0 & 0 & 995 & 0 & 995 & 0 & 0 \\
-1000 & 0 & 0 & 0 & 1000 & 0 & -995 & 0 & 0 & 0 & 995 & 0 \\
0 & 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 992 \\
995 & 0 & 0 & 0 & -995 & 0 & 1000 & 0 & 0 & 0 & -1000 & 0 \\
0 & 995 & 0 & 995 & 0 & 0 & 0 & 1000 & 0 & 1000 & 0 & 0 \\
0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\
0 & 995 & 0 & 995 & 0 & 0 & 0 & 1000 & 0 & 1000 & 0 & 0 \\
-995 & 0 & 0 & 0 & 995 & 0 & -1000 & 0 & 0 & 0 & 1000 & 0 \\
0 & 0 & 0 & 0 & 0 & 992 & 0 & 0 & 0 & 0 & 0 & 1000
\end{bmatrix}$$

For symmetry reasons the rotations around and the translations along the X- and the Y-axes have the same standard deviation. Observe the position of the cameras w.r.t. scene is only 3 to 4 cm. Also, the rotation angles around the Y-and the Y-axis are 0.014 [rad] or appr. 0.8°. Also there are very high correlations between the two sets of pose parameters, some numerically nearly 1.

[7] *b*undle *a*djustment for *c*ameras *s*ystems

If we now determine the relative pose $\mathsf{M} := \mathsf{M}_{12} = \mathsf{M}_1^{-1}\mathsf{M}_2$ we obtain the following set of standard deviations of the twist vector of the relative pose using (18.107) and (18.108)

$$\begin{bmatrix} \sigma_{\rho_1} & \sigma_{\tau_1} \\ \sigma_{\rho_2} & \sigma_{\tau_2} \\ \sigma_{\rho_3} & \sigma_{\tau_3} \end{bmatrix} = \begin{bmatrix} 0.0014 & 0.0028 \\ 0.0014 & 0.0028 \\ 0.0005 & 0.0045 \end{bmatrix} \tag{18.149}$$

The precision of the rotations around and the translations along the X- and Y-axes are approximately 10-times more precise, which is cause by the high correlations of the corresponding pose parameters of the two cameras. The correlation matrix of the relative pose parameters is

$$\begin{bmatrix} 1.0000 & 0 & 0 & 0 & -0.4963 & 0 \\ 0 & 1.0000 & 0 & 0.4963 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0.4963 & 0 & 1.0000 & 0 & 0 \\ -0.4963 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{18.150}$$

showing no correlations above 50 %.

18.7 MATLAB Software

The main routines are available as MATLAB-functions.

1	`calc_A_from_s.m`	$\mathsf{A} = \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}$
2	`calc_concatenated_M_s.m`	$\mathsf{M}(\underline{\boldsymbol{s}}) = \mathsf{M}(\underline{\boldsymbol{s}}_2) \cdot \mathsf{M}(\underline{\boldsymbol{s}}_1)$
4	`calc_concatenated_M_z.m`	$\mathsf{M}(\underline{\boldsymbol{\zeta}}) = \mathsf{M}(\underline{\boldsymbol{\zeta}}_2) \cdot \mathsf{M}(\underline{\boldsymbol{\zeta}}_1)$
5	`calc_concatenated_R.m`	$R(\underline{\boldsymbol{r}}) = R_2(\underline{\boldsymbol{r}}_2) \cdot R(\underline{\boldsymbol{r}}_1)$
6	`calc_inverse_M_s.m`	$\mathsf{M}(\underline{\boldsymbol{s}}^{(-1)}) = \mathsf{M}(\underline{\boldsymbol{s}})$
7	`calc_inverse_M_z.m`	$\mathsf{M}(\underline{\boldsymbol{\zeta}}^{(-1)}) = \mathsf{M}(\underline{\boldsymbol{\zeta}})$
8	`calc_inverse_R.m`	$R(\underline{\boldsymbol{r}}^{(-1)}) = R(\underline{\boldsymbol{r}})$
9	`calc_relative_M_s.m`	$\mathsf{M}(\underline{\boldsymbol{s}}) = \mathsf{M}^{-1}(\underline{\boldsymbol{s}}_1) \cdot \mathsf{M}(\underline{\boldsymbol{s}}_2)$
10	`calc_relative_M_z.m`	$\mathsf{M}(\underline{\boldsymbol{\zeta}}) = \mathsf{M}^{-1}(\underline{\boldsymbol{\zeta}}_1) \cdot \mathsf{M}(\underline{\boldsymbol{\zeta}}_2)$
11	`calc_relative_R.m`	$M(\underline{\boldsymbol{r}}) = R^\mathsf{T}(\underline{\boldsymbol{r}}_1) \cdot R(\underline{\boldsymbol{r}}_2)$
12	`calc_s_from_A.m`	$\mathsf{A} = \begin{bmatrix} S(\boldsymbol{r}) & \boldsymbol{t} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \rightarrow \boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{t} \end{bmatrix}$
13	`calc_z_from_M_M0.m`	$\mathsf{M} = \mathsf{M}(\boldsymbol{\zeta}) \cdot \mathsf{M}_0 \rightarrow \boldsymbol{\zeta}$

Table 18.3: MATLAB routines for rotations and motions in exponential and partially exponential representation

The variables for rotations and motions are structs:

$$\{\texttt{R.R, R.C}\} \quad \{\texttt{M.Ms, M.Cs}\} \quad \{\texttt{M.Mz, M.Cz}\}\,. \tag{18.151}$$

with the covariance matrices `*.C*` having the sizes 3×3, 6×6, and 6×6. For the input of the concatenated and relative rotations and motions we have structs for the transformation pairs:

$$\{\texttt{Rp.Rp, Rp.Cp}\} \quad \{\texttt{Mp.Msp, Mp.Csp}\} \quad \{\texttt{Mp.Mzp, Mp.Czp}\}\,. \tag{18.152}$$

Here the transformations are concatenated leading to

$$\texttt{Rp.Rp} = \texttt{[R1.R,R2.R]} \tag{18.153}$$
$$\texttt{Msp.Msp} = \texttt{[M1s.Ms,M2s.Ms]} \tag{18.154}$$
$$\texttt{Mzp.Mzp} = \texttt{[M1z.Mz,M2z.Mz]}\,. \tag{18.155}$$

The covariance matrices of the pairs allow for correlated transformation parameters, i.e.,

$$\texttt{Rp.Cp} := \begin{bmatrix} \Sigma_{r_1r_1} & \Sigma_{r_1r_2} \\ \Sigma_{r_2r_1} & \Sigma_{r_2r_2} \end{bmatrix} \tag{18.156}$$
$$\texttt{Msp.Csp} := \begin{bmatrix} \Sigma_{s_1s_1} & \Sigma_{s_1s_2} \\ \Sigma_{s_2s_1} & \Sigma_{s_2s_2} \end{bmatrix} \tag{18.157}$$
$$\texttt{Mzp.Czp} := \begin{bmatrix} \Sigma_{\zeta_1\zeta_1} & \Sigma_{\zeta_1\zeta_2} \\ \Sigma_{\zeta_2\zeta_1} & \Sigma_{\zeta_2\zeta_2} \end{bmatrix}. \tag{18.158}$$

In addition we have two routines for each representation to check the implementation:

- `check_basics_rotations.m` and `check_simulated_rotation.m`,
- `check_basics_motion_s.m` and `check_simulated_motion_s.m`, and
- `check_basics_motion_z.m` and `check_simulated_motion_z.m`.

One checks the basic relations:

- vector of adjoint transformtion,
- vector of inverse transformtion,
- function for inverse transformtion,
- vector of concatenated transformtion,
- vector of relative transformtion,
- difference transformation $\mathcal{T}_{12}$ as concatenation of $\mathcal{T}_1^{-1}$ and $\mathcal{T}_2$, hence $\mathcal{T}_{12} = \mathcal{T}_1^{-1} \circ \mathcal{T}_2$.

The output are differences between entities derived in two different manners, which therefore should be numerically small. If no relation fails the numerical test, the transformation is classified as `ok`.

The other checks the whether the mean parameters and their covariance matrix derived from a sample is identical to the given (theoretical) mean and covariance matrix. The output provides the test statistics for the covariance matrix and the mean and the corresponding critical region. E.g. for the exponentially represented motion we obtain:

```
Checks for motions s
Number U of unknown parameters = 6
Redundancy R                   = 6
Number K of samples            = 100
-------------------------------------------------
covariance matrix C_xx  ok: lambda   = 23.9562  in [5.8957,49.0108]
mean of parameters x    ok: mean(dx) =  6.2880  in [0.2994,24.1028]
```

If the prespecified noise standard deviation `sigma_n` is small, generally no test fails. If it is set to `sigma_n`= 0., it is likely that the tests fail due to neglected second order effects. Also, if the number `K` of samples is large, the statistical test becomes more sensitive, such that test statistics may lie outside the critical region.

Finally, the covariance matrices derived with the partially exponential and the exponential representations are compared assuming the motions have been generated with the partially exponential representation. The comparison shows, that the rotations together with their covariance matrix do not significantly differ, but the mean values do:

```
Checks for rotations s|z
Number U of unknown parameters = 3
Redundancy R                   = 3
Number K of samples            = 100
-------------------------------------------------
covariance matrix C_xx  ok: lambda   = 11.8448     in [0.2994,24.1028]
mean of parameters x    ok: mean(dx) =  1.6323     in [0.0153,17.7300]
+++++++++++++++++++++++++++++++++++++++++++++++++
Checks for translations s|z
Number U of unknown parameters = 3
Redundancy R                   = 3
Number K of samples            = 100
-------------------------------------------------
covariance matrix C_xx not ok: lambda   = 108.5578 not in [0.2994,24.1028] *****
mean of parameters x       ok: mean(dx) =   0.5210     in [0.0153,17.7300]
```

18.8 Appendix

18.8.1 Epipolar constraint using motion matrices

If the two images can be modelled as (see Förstner and Wrobel (2016, Eq. (12.34)), PCV)

$$\mathbf{x}' = [\mathsf{K}_1 \mid \mathbf{0}]\mathsf{M}_1^{-1}\mathbf{X} \quad \text{and} \quad \mathbf{x}'' = [\mathsf{K}_2 \mid \mathbf{0}]\mathsf{M}_2^{-1}\mathbf{X}$$

the projection rays are (see PCV (12.76))

$$\mathbf{l}_{\mathrm{x}'} = \mathsf{Q}_1\,\mathbf{L} \quad \text{and} \quad \mathbf{l}_{\mathrm{x}''} = \mathsf{Q}_2\,\mathbf{L}$$

with the projection matrices for lines

$$\mathsf{Q}_1 = [\mathit{0} \mid \mathsf{K}_1^{\circ}]\,\mathsf{M}_{L,1}^{-1} \quad \text{and} \quad \mathsf{Q}_2 = [\mathit{0} \mid \mathsf{K}_2^{\circ}]\,\mathsf{M}_{L,2}^{-1}$$

The motion matrix for lines and its inverse are given by (see PCV (12.75))

$$\mathsf{M}_L = \begin{bmatrix} R & \mathit{0} \\ S(\boldsymbol{Z})R & R \end{bmatrix} \quad \text{and} \quad \mathsf{M}_L^{-1} = \begin{bmatrix} R^{\mathsf{T}} & \mathit{0} \\ R^{\mathsf{T}}S^{\mathsf{T}}(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix}$$

and identical to the adjoint motion matrix, see Table row 5:

$$\boxed{\mathsf{M}_L \equiv \mathrm{Ad}(\mathsf{M})} \tag{18.159}$$

Hence we have the line projection matrices

$$\mathsf{Q}_1 = [\mathit{0} \mid \mathsf{K}_1^{\circ}]\,\mathsf{M}_{\mathrm{ad},1}^{-1} \quad \text{and} \quad \mathsf{Q}_2 = [\mathit{0} \mid \mathsf{K}_2^{\circ}]\,\mathsf{M}_{\mathrm{ad},2}^{-1}$$

Two lines $\mathbf{L}_i, i = 1, 2$ intersect if $\mathbf{L}_1^{\mathsf{T}}\mathsf{D}\mathbf{L}_2 = 0$ (see PCV (7.100)), which is the basis for the definition of the fundamental matrix (see PCV (13.70))

$$\mathsf{F} = \mathsf{Q}_1\mathsf{D}\mathsf{Q}_2^{\mathsf{T}} = [\mathit{0} \mid \mathsf{K}^{\circ}]\,\mathsf{M}_{\mathrm{ad},1}^{-1}\begin{bmatrix} \mathit{0} & \mathit{I} \\ \mathit{I} & \mathit{0} \end{bmatrix}\mathsf{M}_{\mathrm{ad},2}^{-\mathsf{T}}\begin{bmatrix} \mathit{0} \\ \mathsf{K}^{\circ\mathsf{T}} \end{bmatrix}$$

which specializes to the essential matrix assuming the coordinate system in the left image and the motion M from the left to the right camera

$$\boxed{\mathsf{E} = [\mathit{I}_3 \mid \mathit{0}]\mathsf{M}_{\mathrm{ad},2}^{-\mathsf{T}}\begin{bmatrix} \mathit{0} \\ \mathit{I}_3 \end{bmatrix} = S(\boldsymbol{Z})R\,.} \tag{18.160}$$

18.8.2 Differential relation between Euler angles and the exponential representation

A rotation can be represented by Euler angles with the vector

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{18.161}$$

e.g., as

$$R(\boldsymbol{\alpha}) = R_3(\alpha_3)R_2(\alpha_2)R_1(\alpha_1) \tag{18.162}$$

and by a multiplicative representation with a small vector

$$\boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{18.163}$$

as

$$R(\boldsymbol{r}, R^a) := R(\boldsymbol{r})R^a\,. \tag{18.164}$$

The task is to derive the Jacobian

$$J_{r\alpha} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\alpha}}\,. \tag{18.165}$$

We start from the identity of the total derivative

$$\mathrm{d}R = \mathrm{d}R(\boldsymbol{\alpha}) = \mathrm{d}R(\boldsymbol{r}, R^a)\,. \tag{18.166}$$

and aim at finding a relation between $\mathrm{d}\boldsymbol{\alpha}$ and $\mathrm{d}\boldsymbol{r}$ under the assumption $R = R^a$, i.e., differential vectors $\mathrm{d}\boldsymbol{\alpha}$ and $\mathrm{d}\boldsymbol{r}$. We first obtain

$$\begin{aligned} \mathrm{d}R(\boldsymbol{\alpha}) &= \mathrm{d}\left(R_3(\alpha_3)R_2(\alpha_2)R_1(\alpha_1)\right) \qquad (18.167)\\ &= \mathrm{d}R_3(\alpha_3)\left(R_2(\alpha_2)R_1(\alpha_1)\right) + R_3(\alpha_3)\,\mathrm{d}R_2(\alpha_2)\,R_1(\alpha_1) + \left(R_3(\alpha_3)R_2(\alpha_2)\right)\mathrm{d}R_1(\alpha_1) \end{aligned}$$

Now we observe, e.g., for α_1

$$\begin{aligned} \mathrm{d}R_1(\alpha_1) &= \mathrm{d}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_1 & -\sin\alpha_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 \end{bmatrix} && (18.168)\\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin\alpha_1 & -\cos\alpha_1 \\ 0 & \cos\alpha_1 & -\sin\alpha_1 \end{bmatrix}\mathrm{d}\alpha_1 && (18.169)\\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & +1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_1 & -\sin\alpha_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 \end{bmatrix}\mathrm{d}\alpha_1 && (18.170)\\ &= S(\boldsymbol{e}_1)R_1(\alpha_1)\mathrm{d}\alpha_1\,, && (18.171) \end{aligned}$$

or generally

$$\mathrm{d}R_i(\alpha) = S(\boldsymbol{e}_i)R_i(\alpha)\mathrm{d}\alpha_i\,. \tag{18.172}$$

Similarly we thus have

$$\mathrm{d}R_2(\alpha_2) = S(\boldsymbol{e}_2)R_2(\alpha_2)\,\mathrm{d}\alpha_2 \quad \text{and} \quad \mathrm{d}R_3(\alpha_3) = S(\boldsymbol{e}_3)R_3(\alpha_3)\,\mathrm{d}\alpha_3 \tag{18.173}$$

This leads to

$$\begin{aligned} \mathrm{d}R(\boldsymbol{\alpha}) &= S(\boldsymbol{e}_3)\,R_3(\alpha_3)\,R_2(\alpha_2)\,R_1(\alpha_1)\,\mathrm{d}\alpha_3 + && (18.174)\\ &\quad R_3(\alpha_3)\,S(\boldsymbol{e}_2)\,R_2(\alpha_2)\,R_1(\alpha_1)\,\mathrm{d}\alpha_2 + && (18.175)\\ &\quad R_3(\alpha_3)\,R_2(\alpha_2)\,S(\boldsymbol{e}_1)\,R_1(\alpha_1)\,\mathrm{d}\alpha_1 && (18.176) \end{aligned}$$

We now use the relation $R(\boldsymbol{a} \times \boldsymbol{b}) = R\boldsymbol{a} \times R\boldsymbol{b}$ which is valid for all $\boldsymbol{b}$ in the form

$$RS(\boldsymbol{a}) = S(R\boldsymbol{a})R \quad \text{or} \quad RS(\boldsymbol{a})R^{\mathsf{T}} = S(R\boldsymbol{a})\,. \tag{18.177}$$

Then we obtain

$$\begin{aligned} \mathrm{d}R(\boldsymbol{\alpha}) \quad &= \quad S(\boldsymbol{e}_3)R\,\mathrm{d}\alpha_3 + && (18.178)\\ &\quad\; S(R_3(\alpha_3)\boldsymbol{e}_2)R\,\mathrm{d}\alpha_2 + && (18.179)\\ &\quad\; S(R_3(\alpha_3)R_2(\alpha_2)\boldsymbol{e}_1)R\,\mathrm{d}\alpha_1 && (18.180) \end{aligned}$$

or the skew symmetric matrix

$$\begin{aligned} \mathrm{d}R(\boldsymbol{\alpha})R^{\mathsf{T}} \quad &= \quad S(\boldsymbol{e}_3\mathrm{d}\alpha_3) + && (18.181)\\ &\quad\; S(R_3(\alpha_3)\boldsymbol{e}_2\mathrm{d}\alpha_2) + && (18.182)\\ &\quad\; S(R_3(\alpha_3)R_2(\alpha_2)\boldsymbol{e}_1\mathrm{d}\alpha_1) && (18.183) \end{aligned}$$

Now the total differential of $R(\boldsymbol{r}; R^a)$ is given by

$$\mathrm{d}R(\boldsymbol{r}, R^a) = S(\mathrm{d}\boldsymbol{r})R^a \tag{18.184}$$

Hence we have

$$\mathrm{d}R(\boldsymbol{r}, R^a)R^{a\mathsf{T}} = S(\mathrm{d}\boldsymbol{r}) \tag{18.185}$$

Since the approximate rotation matrix is the point of linearization, we have the constraint

$$\mathrm{d}R(\boldsymbol{\alpha})R^{\mathsf{T}} = \mathrm{d}R(\boldsymbol{r}, R^a)R^{\mathsf{T}} \tag{18.186}$$

Therefore the two skew symmetric matrices (18.181) and (18.185) need to be identical. From this we follow

$$\boldsymbol{e}_3\mathrm{d}\alpha_3 + R_3(\alpha_3)\boldsymbol{e}_2\mathrm{d}\alpha_2 + R_3(\alpha_3)R_2(\alpha_2)\boldsymbol{e}_1\mathrm{d}\alpha_1 = \mathrm{d}\boldsymbol{r} \tag{18.187}$$

or

$$\mathrm{d}\boldsymbol{r} = J_{r\alpha}\,\mathrm{d}\boldsymbol{\alpha} \tag{18.188}$$

with the Jacobian

$$\boxed{J_{r\alpha} = [R_3(\alpha_3)R_2(\alpha_2)\boldsymbol{e}_1 \mid R_3(\alpha_3)\boldsymbol{e}_2 \mid \boldsymbol{e}_3]} \tag{18.189}$$

The determinant of the Jacobian is

$$|J_{r\alpha}| = \cos\alpha_2\,. \tag{18.190}$$

This is why for $\cos\alpha_2 = 0$ or for $\alpha_2 = \pm 90°$ there is no unique relation between $\mathrm{d}\boldsymbol{r}$ and $\mathrm{d}\boldsymbol{\alpha}$, which is known as the Gimbal lock.

18.8.3 Adjoint motion matrix in exponential representation

We prove (18.73):

$$\mathrm{Ad}(\mathsf{M}) = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})R & R \end{bmatrix}. \tag{18.191}$$

For this, we express the differential $\mathrm{d}\boldsymbol{s}_{\mathrm{ad}}$ of the small motion vector $\boldsymbol{s}_{\mathrm{ad}}$ directly as a function of the differential $\mathrm{d}\boldsymbol{s}$. We start from (18.71)

$$\mathrm{d}\mathsf{M}(\boldsymbol{s}_{\mathrm{ad}})\,\mathsf{M} = \mathsf{M}\,\mathrm{d}\mathsf{M}(\boldsymbol{s}) \tag{18.192}$$

with its differential

$$\mathsf{A}(\mathrm{d}\boldsymbol{s}_{\mathrm{ad}})\,\mathsf{M} = \mathsf{M}\mathsf{A}(\mathrm{d}\boldsymbol{s}) \tag{18.193}$$

With the vector

$$\mathrm{d}\boldsymbol{s}_{\mathrm{ad}} = \begin{bmatrix} \mathrm{d}\boldsymbol{r}_{\mathrm{ad}} \\ \mathrm{d}\boldsymbol{t}_{\mathrm{ad}} \end{bmatrix} \tag{18.194}$$

this explicitly yields

$$\begin{bmatrix} S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}}) & \mathrm{d}\boldsymbol{t}_{\mathrm{ad}} \\ \mathbf{T} & 0 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} S(\mathrm{d}\boldsymbol{r}) & \mathrm{d}\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \tag{18.195}$$

$$\begin{bmatrix} S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}})\, R & S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}})\, \boldsymbol{Z} + \mathrm{d}\boldsymbol{t}_{\mathrm{ad}} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} = \begin{bmatrix} R\, S(\mathrm{d}\boldsymbol{r}) & R\mathrm{d}\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \tag{18.196}$$

$$\begin{bmatrix} S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}})\, R & S(\mathrm{d}\boldsymbol{r}_{\mathrm{ad}})\, \boldsymbol{Z} + \mathrm{d}\boldsymbol{t}_{\mathrm{ad}} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} = \begin{bmatrix} S(R\mathrm{d}\boldsymbol{r})\, R & R\mathrm{d}\boldsymbol{t} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \tag{18.197}$$

hence by comparing the upper left submatrices

$$\mathrm{d}\boldsymbol{r}_{\mathrm{ad}} = R\, \mathrm{d}\boldsymbol{r}\,, \tag{18.198}$$

and therefore

$$\mathrm{d}\boldsymbol{t}_{\mathrm{ad}} = R\, \mathrm{d}\boldsymbol{t} + S(\boldsymbol{Z})\mathrm{d}\boldsymbol{r}_{\mathrm{ad}}\,. \tag{18.199}$$

Compound this reads as

$$\boxed{\mathrm{d}\boldsymbol{s}_{\mathrm{ad}} = \mathrm{Ad}(\mathsf{M})\, \mathrm{d}\boldsymbol{s} \quad \text{with} \quad \mathrm{Ad}(\mathsf{M}) = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})R & R \end{bmatrix}.} \tag{18.200}$$

with the adjoint motion matrix $\mathrm{Ad}(\mathsf{M})$.

18.8.4 Adjoint motion in partially exponential representation

We prove (18.94)

$$\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}} = {}^{\zeta}\mathsf{M}_{\mathrm{ad}}\, \mathrm{d}\boldsymbol{\zeta} \quad \text{with} \quad {}^{\zeta}\mathsf{M}_{\mathrm{ad}} = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})\, R & R \end{bmatrix} \tag{18.201}$$

with the vector

$$\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}} = \begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_{\mathrm{ad}} \\ \mathrm{d}\boldsymbol{\tau}_{\mathrm{ad}} \end{bmatrix}. \tag{18.202}$$

We have start from

$$\mathsf{M}(\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}}) = \mathsf{M}\, \mathsf{M}(\mathrm{d}\boldsymbol{\zeta})\ \mathsf{M}^{-1} \tag{18.203}$$

The differential reads

$$\begin{aligned} \begin{bmatrix} S(\mathrm{d}\boldsymbol{\rho}_{\mathrm{ad}}) & \mathrm{d}\boldsymbol{\tau}_{\mathrm{ad}} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} &= \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} S(\mathrm{d}\boldsymbol{\rho}) & \mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \\ &= \begin{bmatrix} R\, S(\mathrm{d}\boldsymbol{r}) & R\mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} && (18.204) \\ &= \begin{bmatrix} RS(\mathrm{d}\boldsymbol{\rho})R^{\mathsf{T}} & -RS(\mathrm{d}\boldsymbol{\rho})R^{\mathsf{T}}\boldsymbol{Z} + R\mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} && (18.205) \\ &= \begin{bmatrix} S(R\mathrm{d}\boldsymbol{\rho}) & -S(R\mathrm{d}\boldsymbol{\rho})\boldsymbol{Z} + R\mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} && (18.206) \\ &= \begin{bmatrix} S(R\mathrm{d}\boldsymbol{\rho}) & S(\boldsymbol{Z})R\mathrm{d}\boldsymbol{\rho} + R\mathrm{d}\boldsymbol{\tau} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix} && (18.207) \end{aligned}$$

From the upper left sub-matrix we conclude

$$\mathrm{d}\boldsymbol{\rho}_{\mathrm{ad}} = R\, \mathrm{d}\boldsymbol{\rho}\,. \tag{18.208}$$

With this relation we obtain from the upper right part

$$\mathrm{d}\boldsymbol{\tau}_{\mathrm{ad}} = R\, \mathrm{d}\boldsymbol{\tau} + S(\boldsymbol{Z})R\, \mathrm{d}\boldsymbol{\rho}\,. \tag{18.209}$$

Joined this can be written as

$$\boxed{\mathrm{d}\boldsymbol{\zeta}_{\mathrm{ad}} = {}^{\zeta}\mathsf{M}_{\mathrm{ad}}\ \mathrm{d}\boldsymbol{\zeta} \quad \text{with} \quad {}^{\zeta}\mathsf{M}_{\mathrm{ad}} = \begin{bmatrix} R & 0 \\ S(\boldsymbol{Z})\ R & R \end{bmatrix}}. \tag{18.210}$$

with the adjoint motion matrix ${}^{\zeta}\mathsf{M}_{\mathrm{ad}}$, which is the same as for the exponential representation.

18.8.5 Uncertain inverse rotation

We prove (18.36). We have the relation

$$R(\underline{\boldsymbol{r}}^{(-1)})\ R^{-1} = (R(\underline{\boldsymbol{r}})\ R)^{-1} = R^{\mathsf{T}} R^{\mathsf{T}}(\underline{\boldsymbol{r}})\,. \tag{18.211}$$

Taking the total differential we obtain

$$S(\mathrm{d}\boldsymbol{r}^{(-1)})\ R^{-1} = R^{\mathsf{T}} S^{\mathsf{T}}(\mathrm{d}\boldsymbol{r})\,. \tag{18.212}$$

This yields

$$S(\mathrm{d}\boldsymbol{r}^{(-1)}) = R^{\mathsf{T}} S^{\mathsf{T}}(\mathrm{d}\boldsymbol{r}) R = S(-R^{\mathsf{T}}\mathrm{d}\boldsymbol{r})\,. \tag{18.213}$$

Thus we obtain the Jacobian

$$\boxed{J_{r^{(-1)}r} = \frac{\partial \boldsymbol{r}^{(-1)}}{\partial \boldsymbol{r}} = -R^{\mathsf{T}}\,.} \tag{18.214}$$

Remark: If we would have defined the uncertain rotation with a noisy rotation from the right $\underline{R} = R R(\underline{\boldsymbol{r}})$, we would have obtained:

$$R^{-1}\ R(\underline{\boldsymbol{r}}^{(-1)}) = (R\ R(\underline{\boldsymbol{r}}))^{-1} = R^{\mathsf{T}}(\underline{\boldsymbol{r}})\ R^{\mathsf{T}}\,, \tag{18.215}$$

thus the differential

$$S(\mathrm{d}\boldsymbol{r}^{(-1)}) = R\ S^{\mathsf{T}}(\mathrm{d}\boldsymbol{r})\ R^{\mathsf{T}} = S^{\mathsf{T}}(R\mathrm{d}\boldsymbol{r})\,, \tag{18.216}$$

thus

$$\mathrm{d}\boldsymbol{r}^{(-1)} = -R_{\mathrm{ad}}\ \mathrm{d}\boldsymbol{r} = -R\ \mathrm{d}\boldsymbol{r}\,. \tag{18.217}$$

This relation is slightly more intuitive than (18.214). ◇

18.8.6 Uncertain inverse motion in exponential representation

We prove (18.78)

$$\mathrm{d}\boldsymbol{s}^{(-1)} = -\mathrm{Ad}(\mathsf{M})^{-1}\ \mathrm{d}\boldsymbol{s}\,. \tag{18.218}$$

We have the basic relation

$$\exp(\mathsf{A}(\boldsymbol{s}^{(-1)})) \cdot \mathsf{M}^{-1} = (\exp(\mathsf{A}(\boldsymbol{s})) \cdot \mathsf{M})^{-1} = \mathsf{M}^{-1}(\exp(\mathsf{A}(\boldsymbol{s}))^{-1}\,. \tag{18.219}$$

Taking the total differential, and using the first order approximation of $(\exp(X))^{-1} = I - X + 1/2X^2 - \ldots$ we obtain by taking the total differential

$$\mathsf{A}\left(\mathrm{d}\boldsymbol{s}^{(-1)}\right)\ \mathsf{M}^{-1} = -\mathsf{M}^{-1}\ \mathsf{A}(\mathrm{d}\boldsymbol{s})\,. \tag{18.220}$$

This yields

$$\mathsf{A}\left(\mathrm{d}\boldsymbol{s}^{(-1)}\right) = -\mathsf{M}^{-1}\ \mathsf{A}(\mathrm{d}\boldsymbol{s})\mathsf{M}\,, \tag{18.221}$$

or

$$\mathsf{M}\mathsf{A}\left(\mathrm{d}\boldsymbol{s}^{(-1)}\right)\mathsf{M}^{-1} = -\mathsf{A}(\mathrm{d}\boldsymbol{s}) \tag{18.222}$$

thus using (18.75)

$$\mathsf{A}\left(\mathrm{Ad}(\mathsf{M})\mathrm{d}\boldsymbol{s}^{(-1)}\right) = -\mathsf{A}(\mathrm{d}\boldsymbol{s}) \tag{18.223}$$

Therefore, we obtain the Jacobian

$$J_{s^{(-1)}s} = \frac{\partial \boldsymbol{s}^{(-1)}}{\partial \boldsymbol{s}} = -\mathsf{M}_{\mathrm{ad}}^{-1}\,, \tag{18.224}$$

which yields

$$\boxed{\mathrm{d}\boldsymbol{s}^{(-1)} = -\mathrm{Ad}(\mathsf{M})^{-1}\ \mathrm{d}\boldsymbol{s}\,.} \tag{18.225}$$

18.8.7 Uncertain inverse motion in partially exponential representation

We prove (18.97)

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}^{(-1)} \end{bmatrix} = -\begin{bmatrix} R^{\mathsf{T}} & \boldsymbol{0} \\ -R^{\mathsf{T}}S^{\mathsf{T}}(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix}. \tag{18.226}$$

It should hold

$$\begin{bmatrix} R(\underline{\boldsymbol{\rho}})R & \boldsymbol{Z}+\underline{\boldsymbol{\tau}} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} R(\underline{\boldsymbol{\rho}}^{(-1)})R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{Z}+\underline{\boldsymbol{\tau}}^{(-1)} \\ \boldsymbol{0} & 1 \end{bmatrix} = I_4\,, \tag{18.227}$$

or

$$\begin{bmatrix} R(\underline{\boldsymbol{\rho}})RR(\underline{\boldsymbol{\rho}}^{(-1)})R^{\mathsf{T}} & R(\underline{\boldsymbol{\rho}})R(-R^{\mathsf{T}}\boldsymbol{Z}+\underline{\boldsymbol{\tau}}^{(-1)})+\boldsymbol{Z}+\underline{\boldsymbol{\tau}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = I_4\,. \tag{18.228}$$

The differential of the upper left submatrix is

$$S(\mathrm{d}\boldsymbol{\rho}) + RR(\mathrm{d}\boldsymbol{\rho}^{(-1)})R^{\mathsf{T}} = S(\mathrm{d}\boldsymbol{\rho}) + R(R\mathrm{d}\boldsymbol{\rho}^{(-1)}) = \boldsymbol{0}\,. \tag{18.229}$$

Therefore we obtain

$$\mathrm{d}\boldsymbol{\rho}^{(-1)} = -R^{\mathsf{T}}\,\mathrm{d}\boldsymbol{\rho}\,. \tag{18.230}$$

The differential of the upper right matrix is

$$-S(\mathrm{d}\boldsymbol{\rho})\boldsymbol{Z} + R\mathrm{d}\boldsymbol{\tau}^{(-1)} + \mathrm{d}\boldsymbol{\tau} = \boldsymbol{0} \tag{18.231}$$

This yields

$$\mathrm{d}\boldsymbol{\tau}^{(-1)} = R^{\mathsf{T}}S(\mathrm{d}\boldsymbol{\rho})\boldsymbol{Z} - R^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau} = -R^{\mathsf{T}}S(\boldsymbol{Z})\mathrm{d}\boldsymbol{\rho} - R^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}\,. \tag{18.232}$$

This can be written as

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}^{(-1)} \end{bmatrix} = -\begin{bmatrix} R^{\mathsf{T}} & \boldsymbol{0} \\ R^{\mathsf{T}}S(\boldsymbol{Z}) & R^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix} \tag{18.233}$$

18.8.8 Uncertain concatenated motions in exponential representation

We prove (18.82)

$$\mathrm{d}\boldsymbol{s} = \mathsf{M}_2\mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2\,. \tag{18.234}$$

We start from the total differential of $\underline{\mathsf{M}} = \underline{\mathsf{M}}_2\underline{\mathsf{M}}_1$:

$$\mathsf{A}(\underline{\boldsymbol{s}})\mathsf{M} = \mathsf{A}(\underline{\boldsymbol{s}}_2)\mathsf{M} + \mathsf{M}_2\mathsf{A}(\underline{\boldsymbol{s}}_1)\mathsf{M}_1 \tag{18.235}$$

or multiplying with $\mathsf{M}^{-1} = \mathsf{M}_1^{-1}\mathsf{M}_2^{-1}$ from the right

$$\mathsf{A}(\underline{\boldsymbol{s}}) = \mathsf{M}_2\mathsf{A}(\underline{\boldsymbol{s}}_1)\mathsf{M}_2^{-1} + \mathsf{A}(\underline{\boldsymbol{s}}_2)\,. \tag{18.236}$$

With (18.75) we thus obtain

$$\mathsf{A}(\underline{\boldsymbol{s}}) = \mathsf{A}(\mathsf{M}_{2\mathrm{ad}}\ \underline{\boldsymbol{s}}_1) + \mathsf{A}(\underline{\boldsymbol{s}}_2)\,. \tag{18.237}$$

This allows to express the differential motion parameters as

$$\mathrm{d}\boldsymbol{s} = \mathsf{M}_{2\mathrm{ad}}\mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2\,. \tag{18.238}$$

18.8.9 Uncertain concatenated motions in partially exponential representation

We prove (18.102)

$$\mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{\rho}_2 + \mathrm{d}R_2\boldsymbol{\rho}_1 \quad \text{and} \quad \mathrm{d}\boldsymbol{\tau} = \mathrm{d}\boldsymbol{\tau}_2 + R_2\mathrm{d}\boldsymbol{\tau}_1 - S(R_2\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_2\,. \tag{18.239}$$

We explicitly have

$$\underline{\mathsf{M}} \;=\; \begin{bmatrix} R(\underline{\boldsymbol{\rho}})R & \boldsymbol{Z}+\underline{\boldsymbol{\tau}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} R(\underline{\boldsymbol{\rho}}) & (I_3 - R(\underline{\boldsymbol{\rho}}))\boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{Z} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.240}$$

and similarly

$$\underline{\mathsf{M}}_i \;=\; \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_i)R_i & \boldsymbol{Z}_i+\underline{\boldsymbol{\tau}}_i \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$$

Therefore

$$\underline{\mathsf{M}} \;=\; \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_2)R_2 & \boldsymbol{Z}_2+\underline{\boldsymbol{\tau}}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_1)R_1 & \boldsymbol{Z}_1+\underline{\boldsymbol{\tau}}_1 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.241}$$

$$= \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_2)R_2R(\underline{\boldsymbol{\rho}}_1)R_1 & R(\underline{\boldsymbol{\rho}}_2)R_2(\boldsymbol{Z}_1+\underline{\boldsymbol{\tau}}_1)+\boldsymbol{Z}_2+\underline{\boldsymbol{\tau}}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.242}$$

We now linearize, multiplicatively for R, additively for $\boldsymbol{Z}$:

$$\begin{bmatrix} \left(S(\mathrm{d}\boldsymbol{\rho}_2) + R_2S(\mathrm{d}\boldsymbol{\rho}_1)R_2^{\mathsf{T}}\right) R & (R_2\boldsymbol{Z}_1+\boldsymbol{Z}_2) + S(\mathrm{d}\boldsymbol{\rho}_2)R_2\boldsymbol{Z}_1 + R_2\mathrm{d}\boldsymbol{\tau}_1 + \mathrm{d}\boldsymbol{\tau}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (S(\mathrm{d}\boldsymbol{\rho}_2 + R_2\mathrm{d}\boldsymbol{\rho}_1))\;R & \boldsymbol{Z} + S(\mathrm{d}\boldsymbol{\rho}_2)R_2\boldsymbol{Z}_1 + R_2\mathrm{d}\boldsymbol{\tau}_1 + \mathrm{d}\boldsymbol{\tau}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.243}$$

By comparison with (18.240) we find

$$\mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{\rho}_2 + \mathrm{d}R_2\boldsymbol{\rho}_1 \quad \text{and} \quad \mathrm{d}\boldsymbol{\tau} = \mathrm{d}\boldsymbol{\tau}_2 + R_2\mathrm{d}\boldsymbol{\tau}_1 - S(R_2\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_2 \tag{18.244}$$

Relation to the concatenated motion with exponential representation. We can write (18.244) as

$$\mathrm{d}\boldsymbol{\zeta} = \mathsf{M}_{\mathrm{con1}}\,\mathrm{d}\boldsymbol{\zeta}_1 + \mathsf{M}_{\mathrm{con2}}\mathrm{d}\boldsymbol{\zeta}_2\,, \tag{18.245}$$

with

$$\mathsf{M}_{\mathrm{con1}} = \begin{bmatrix} R_2 & 0 \\ 0 & R_2 \end{bmatrix} \quad \text{and} \quad \mathsf{M}_{\mathrm{con2}} = \begin{bmatrix} I_3 & 0 \\ -S(R_2\boldsymbol{Z}_1) & I_3 \end{bmatrix}. \tag{18.246}$$

Using the Jacobians $J_{s\zeta}$ for switching between the representations, see (18.67) we can show, that this leads to the form

$$\mathrm{d}\boldsymbol{s} = \mathsf{M}_{2\mathrm{ad}}\,\mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{\zeta}_2 \quad \text{with} \quad \mathsf{M}_{2\mathrm{ad}} = \begin{bmatrix} R_2 & 0 \\ S(\boldsymbol{Z}_2)R_2 & R_2 \end{bmatrix}. \tag{18.247}$$

In detail we have

$$\mathrm{d}\boldsymbol{\zeta} \;=\; \mathsf{M}_{\mathrm{con1}}\,\mathrm{d}\boldsymbol{\zeta}_1 + \mathsf{M}_{\mathrm{con2}}\mathrm{d}\boldsymbol{\zeta}_2 \tag{18.248}$$

$$J_{\zeta s}\mathrm{d}\boldsymbol{s} \;=\; \mathsf{M}_{\mathrm{con1}}\,J_{1,\zeta s}\mathrm{d}\boldsymbol{s}_1 + \mathsf{M}_{\mathrm{con2}}J_{2,\zeta s}\mathrm{d}\boldsymbol{s}_2 \tag{18.249}$$

$$\mathrm{d}\boldsymbol{s} \;=\; J_{\zeta s}^{-1}\mathsf{M}_{\mathrm{con1}}\,J_{1,\zeta s}\mathrm{d}\boldsymbol{s}_1 + J_{\zeta s}^{-1}\mathsf{M}_{\mathrm{con2}}J_{2,\zeta s}\mathrm{d}\boldsymbol{s}_2 \tag{18.250}$$

Now we use

$$J_{\zeta s} = \begin{bmatrix} I_3 & 0 \\ -S(R_2\boldsymbol{Z}_1+\boldsymbol{Z}_2) & I_3 \end{bmatrix} \quad \text{and} \quad J_{i,\zeta s} = \begin{bmatrix} I_3 & 0 \\ -S(\boldsymbol{Z}_i) & I_3 \end{bmatrix}$$

and first obtain

$$J_{\zeta s}^{-1}\mathsf{M}_{\text{con1}}\, J_{1,\zeta s} = \begin{bmatrix} R_2 & 0 \\ S(R_2\boldsymbol{Z}_1 + \boldsymbol{Z}_2)R_2 & R_2 \end{bmatrix} J_{1,\zeta s} \tag{18.251}$$

$$= \begin{bmatrix} R_2 & 0 \\ S(R_2\boldsymbol{Z}_1 + \boldsymbol{Z}_2)R_2 - R_2S(\boldsymbol{Z}_1) & R_2 \end{bmatrix} \tag{18.252}$$

$$= \begin{bmatrix} R_2 & 0 \\ S(\boldsymbol{Z}_2)R_2 + \underbrace{S(R_2\boldsymbol{Z}_1)R_2 - R_2S(\boldsymbol{Z}_1)}_{=0} & R_2 \end{bmatrix} \tag{18.253}$$

$$= \begin{bmatrix} R_2 & 0 \\ S(\boldsymbol{Z}_2)R_2 & R_2 \end{bmatrix}. \tag{18.254}$$

Similarly we have

$$J_{\zeta s}^{-1}\mathsf{M}_{\text{con2}}J_{2,\zeta s} = \begin{bmatrix} I_3 & 0 \\ S(R_2\boldsymbol{Z}_1 + \boldsymbol{Z}_2) - S(R_2\boldsymbol{Z}_1) & I_3 \end{bmatrix} J_{2,\zeta s} = \begin{bmatrix} I_3 & 0 \\ 0 & I_3 \end{bmatrix},$$

which yields

$$\mathrm{d}\boldsymbol{s} = \mathsf{M}_{2\text{ad}}\, \mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2\,. \tag{18.255}$$

18.8.10 Uncertain relative motion in exponential representation

We prove (18.87)

$$\mathrm{d}\boldsymbol{s}_{12} = \mathsf{M}_{1,\text{ad}}^{-1}\,(\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1)\,. \tag{18.256}$$

The uncertain relative motion is

$$\mathsf{M}(\underline{\boldsymbol{s}}_{12})\,\mathsf{M}_{12} = (\mathsf{M}(\underline{\boldsymbol{s}}_1)\,\mathsf{M}_1)^{-1}\mathsf{M}(\underline{\boldsymbol{s}}_2)\,\mathsf{M}_2\,., \tag{18.257}$$

or

$$\mathsf{M}(\mathrm{d}\boldsymbol{s}_{12})\,\mathsf{M}_{12} = \mathsf{M}_1^{-1}\,\mathsf{M}^{-1}(\mathrm{d}\boldsymbol{s}_1)\mathsf{M}(\mathrm{d}\boldsymbol{s}_2)\,\mathsf{M}_2\,. \tag{18.258}$$

Taking the total differential, we obtain

$$\mathsf{A}(\mathrm{d}\boldsymbol{s}_{12})\,\mathsf{M}_{12} = \mathsf{M}_1^{-1}\,\mathsf{A}(-\mathrm{d}\boldsymbol{s}_1)\,\mathsf{M}_2 + \mathsf{M}_1^{-1}\,\mathsf{A}(\mathrm{d}\boldsymbol{s}_2)\,\mathsf{M}_2\,. \tag{18.259}$$

or

$$\mathsf{A}(\mathrm{d}\boldsymbol{s}_{12}) = \mathsf{M}_1^{-1}\,\mathsf{A}(\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1)\,\mathsf{M}_1 \tag{18.260}$$

or

$$\mathsf{M}_1\mathsf{A}(\mathrm{d}\boldsymbol{s}_{12})\mathsf{M}_1^{-1} = \mathsf{A}(\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1) \tag{18.261}$$

Hence, with

$$\mathsf{M}_1\mathsf{A}(\mathrm{d}\boldsymbol{s}_{12})\mathsf{M}_1^{-1} = \mathsf{A}(\mathrm{Ad}(\mathsf{M}_1)\,\mathrm{d}\boldsymbol{s}_{12}) \tag{18.262}$$

Therefore we finally have the relation

$$\mathrm{d}\boldsymbol{s}_{12} = \mathsf{M}_{1,\text{ad}}^{-1}\,(\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1)\,. \tag{18.263}$$

Check using the inverse and the concatenation We start from the concatenation

$$\mathsf{M} = \mathsf{M}_2\mathsf{M}_1\,, \tag{18.264}$$

use (18.78) and (18.238)

$$\mathrm{d}\boldsymbol{s}^{(-1)} = -\mathsf{M}_{\text{ad}}^{-1}\,\mathrm{d}\boldsymbol{s} \quad \text{and} \quad \mathrm{d}\boldsymbol{s} = \mathrm{Ad}(\mathsf{M}_2)\,\mathrm{d}\boldsymbol{s}_1 + \mathrm{d}\boldsymbol{s}_2\,, \tag{18.265}$$

and apply this to

$$\mathsf{M}_{12} = \mathsf{M}_1^{-1}\mathsf{M}_2\,. \tag{18.266}$$

This yields

$$\mathrm{d}\boldsymbol{s}_{12} = \mathsf{M}_{1,\text{ad}}^{-1}\,\mathrm{d}\boldsymbol{s}_2 - \mathsf{M}_{1,\text{ad}}^{-1}\,\mathrm{d}\boldsymbol{s}_1 = \mathsf{M}_{1,\text{ad}}^{-1}\,(\mathrm{d}\boldsymbol{s}_2 - \mathrm{d}\boldsymbol{s}_1)\,. \tag{18.267}$$

18.8.11 Uncertain relative motion in partially exponential representation

We prove (18.107) and (18.108)

$$\mathrm{d}\boldsymbol{\rho}_{12} = R_1^{\mathsf{T}}\mathrm{d}(\boldsymbol{\rho}_2 - \boldsymbol{\rho}_1) \quad \text{and} \quad \mathrm{d}\boldsymbol{\tau}_{12} = R_1^{\mathsf{T}}S(\boldsymbol{Z}_2 - \boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1 + R_1^{\mathsf{T}}\mathrm{d}(\boldsymbol{\tau}_2 - \boldsymbol{\tau}_1)\,. \tag{18.268}$$

We start from

$$\underline{\mathsf{M}}_i := \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_i)R_i & Z_i + \underline{\boldsymbol{\tau}}_i \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.269}$$

and obtain

$$\begin{aligned} \underline{\mathsf{M}}_{12} &= \underline{\mathsf{M}}_1^{-1}\,\underline{\mathsf{M}}_2 && (18.270)\\ &= \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_1)R_1 & Z_1 + \underline{\boldsymbol{\tau}}_1 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}^{-1} \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_2)R_2 & Z_2 + \underline{\boldsymbol{\tau}}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} && (18.271)\\ &= \begin{bmatrix} R_1^{\mathsf{T}}R^{\mathsf{T}}(\underline{\boldsymbol{\rho}}_1) & -(R_1^{\mathsf{T}}R^{\mathsf{T}}(\underline{\boldsymbol{\rho}}_1))(Z_1 + \underline{\boldsymbol{\tau}}_1) \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_2)R_2 & Z_2 + \underline{\boldsymbol{\tau}}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_1^{\mathsf{T}}R^{\mathsf{T}}(\underline{\boldsymbol{\rho}}_1)R(\underline{\boldsymbol{\rho}}_2)R_2 & R_1^{\mathsf{T}}R^{\mathsf{T}}(\underline{\boldsymbol{\rho}}_1)(Z_2 + \underline{\boldsymbol{\tau}}_2) - (R_1^{\mathsf{T}}R^{\mathsf{T}}(\underline{\boldsymbol{\rho}}_1))(Z_1 + \underline{\boldsymbol{\tau}}_1) \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \end{aligned}$$

Linearizing the rotation multiplicatively and the translation additively we have

$$\begin{aligned} \mathsf{M}_{12} &\approx \begin{bmatrix} R_1^{\mathsf{T}}(S^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_1) + S(\mathrm{d}\boldsymbol{\rho}_2))R_2 & R_1^{\mathsf{T}}S^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_1)\boldsymbol{Z}_2 + R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}_2) - R_1^{\mathsf{T}}S^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_1))\boldsymbol{Z}_1 - R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}_1) \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_1^{\mathsf{T}}S(\mathrm{d}\boldsymbol{\rho}_2 - \mathrm{d}\boldsymbol{\rho}_2) & R_1^{\mathsf{T}}S(\boldsymbol{Z}_2 - \boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1(Z_2 + \mathrm{d}\boldsymbol{\tau}_2) + R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{\tau}_2 - \mathrm{d}\boldsymbol{\tau}_1) \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \end{aligned} \tag{18.272}$$

Check using the inverse and the concatenation We start from the concatenation

$$\mathsf{M} = \mathsf{M}_l\mathsf{M}_r\,, \tag{18.273}$$

use (18.97) and (18.102)

$$\mathrm{d}\boldsymbol{\zeta}^{(-1)} = \begin{bmatrix} \mathrm{d}\boldsymbol{\rho}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}^{(-1)} \end{bmatrix} = \begin{bmatrix} -R^{\mathsf{T}} & 0 \\ -R^{\mathsf{T}}S(\boldsymbol{Z}) & -R^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix} \tag{18.274}$$

and

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho} \\ \mathrm{d}\boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} R_l\mathrm{d}\boldsymbol{\rho}_r + \mathrm{d}\boldsymbol{\rho}_l \\ R_l\mathrm{d}\boldsymbol{\tau}_r + \mathrm{d}\boldsymbol{\tau}_l - S(R_l\boldsymbol{Z}_r)\mathrm{d}\boldsymbol{\rho}_l \end{bmatrix}. \tag{18.275}$$

and apply this to

$$\mathsf{M}_{lr} := \underbrace{\mathsf{M}_1^{-1}}_{\mathsf{M}_l}\,\underbrace{\mathsf{M}_2}_{\mathsf{M}_r} = \begin{bmatrix} R_1^{\mathsf{T}} & -R_1^{\mathsf{T}}\boldsymbol{Z}_1 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} R_2 & \boldsymbol{Z}_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \underbrace{R_1^{\mathsf{T}}R_2}_{R_{12}} & \underbrace{R_1^{\mathsf{T}}(\boldsymbol{Z}_2 - \boldsymbol{Z}_1)}_{\boldsymbol{Z}_{12}} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{18.276}$$

We obtain

$$\underline{\mathsf{M}}_{lr} := \begin{bmatrix} R(\underline{\boldsymbol{\rho}}_{12})R_{12} & Z_{12} + \underline{\boldsymbol{\tau}}_{12} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{18.277}$$

We use

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_r \\ \mathrm{d}\boldsymbol{\tau}_r \end{bmatrix} := \begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_2 \\ \mathrm{d}\boldsymbol{\tau}_2 \end{bmatrix} \tag{18.278}$$

and

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_l \\ \mathrm{d}\boldsymbol{\tau}_l \end{bmatrix} := \begin{bmatrix} -R_1^{\mathsf{T}} & 0 \\ -R_1^{\mathsf{T}}S(\boldsymbol{Z}_1) & -R_1^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_1 \\ \mathrm{d}\boldsymbol{\tau}_1 \end{bmatrix} = \begin{bmatrix} -R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\rho}_1 \\ -R_1^{\mathsf{T}}S(\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1 - R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}_1 \end{bmatrix} \tag{18.279}$$

Now, we have

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_{lr} \\ \mathrm{d}\boldsymbol{\tau}_{lr} \end{bmatrix} = \begin{bmatrix} \mathrm{d}\boldsymbol{\rho}_{12}^{(-1)} \\ \mathrm{d}\boldsymbol{\tau}_{12}^{(-1)} \end{bmatrix} \tag{18.280}$$

$$= \begin{bmatrix} R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\rho}_2 - R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\rho}_1 \\ R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}_2 - R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\tau}_1 - R_1^{\mathsf{T}}S(\boldsymbol{Z}_1)\mathrm{d}\boldsymbol{\rho}_1 - S(R_1^{\mathsf{T}}\boldsymbol{Z}_2)(-R_1^{\mathsf{T}}\mathrm{d}\boldsymbol{\rho}_1) \end{bmatrix}$$

$$= \begin{bmatrix} R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{\rho}_2 - \mathrm{d}\boldsymbol{\rho}_1) \\ R_1^{\mathsf{T}}(\mathrm{d}\boldsymbol{\tau}_2 - \mathrm{d}\boldsymbol{\tau}_1) + R_1^{\mathsf{T}}S(-\boldsymbol{Z}_1 + \boldsymbol{Z}_2)\mathrm{d}\boldsymbol{\rho}_1 \end{bmatrix} \tag{18.281}$$

19 Centroid Form of an Uncertain Plane

A plane can be represented in various manners. We especially discuss the centroid form of an uncertain plane, which naturally results from estimating a plane from a given point set. We discuss the representation, its recursive estimation assuming isotropic point uncertainty and optimal estimation.

19.1 Problem

This note (2020) collects methods for representing and estimating uncertain planes. It focusses on the geometrically intuitive centroid representation, naturally resulting from fitting a plane through a point cloud. We collect methods for estimating a plane from scene points, for averaging uncertain planes and for estimating a motion for plane correspondences.

The statistically rigorous estimation, discussed here in Sect. 19.3.2, has the advantage of giving insight into the uncertainty structure, whereas the solution based on spherically normalized homogeneous plane coordinates in Note 20 is technically more elegant, and easily generalizes to the estimation of multiple planes.

A natural representation of an uncertain plane is its centroid form

$$\mathcal{A}: \quad \{\boldsymbol{X}_0, Q; \sigma_q, \sigma_\phi, \sigma_\psi\} , \tag{19.1}$$

see Fig. 19.1. This representation can directly be derived from a set of 3D points $\mathcal{X}_i, i = 1, ..., I$ with isotropic uncertainty $\Sigma_{X_iX_i} = \sigma_i^2 I_3$.

This note addresses three problems, namely

1. the estimation of a plane from uncertain points,
2. the estimation of a spatial motion from plane-to-point correspondences, and
3. the estimation of a spatial motion from plane-to-plane correspondences.

19.2 Centroid Representation of a Plane

19.2.1 The Representation

The centroid representation of a plane is given by (see Fig. 19.1)

$$\boxed{\mathcal{A}: \quad \{\boldsymbol{X}_0, Q; \sigma_q, \sigma_\phi, \sigma_\psi\}\ .} \tag{19.2}$$

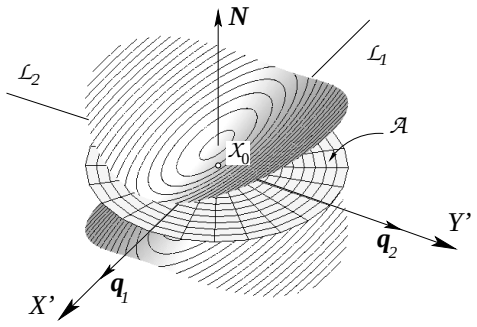


Figure 19.1: Uncertain plane $\mathcal{A}$. Its center is $\mathcal{X}_0$; the center is that point of the plane where the uncertainty across the (perpendicular to the) plane is smallest; it is uncertain along the normal by σ_q. Its normal is $\boldsymbol{N}$; its rotational uncertainty is composed of two independent uncertain rotations around $\mathcal{L}_1$ and $\mathcal{L}_2$ which are mutually perpendicular. The standard deviations σ_ϕ and σ_ψ are the uncertainties of the X'- and Y'-components of the normal $\boldsymbol{N}$. The three directions form an orthonormal tripod $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{N}]$

Here we have:

- the coordinates of the centroid $\mathcal{Z}$;
- the rotation matrix
$$Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3] \tag{19.3}$$
with its normal
$$\boldsymbol{N} = \boldsymbol{q}_3 = Q\boldsymbol{e}_3 \tag{19.4}$$
and the local coordinate system $[\boldsymbol{q}_1, \boldsymbol{q}_2]$ in the plane, where $\boldsymbol{q}_1$ is the major axis, and $\boldsymbol{q}_2$ is the minor axis of the moment matrix point cloud, when projected into the plane.
- the variance σ_q^2 across the plane;
- the variances σ_ϕ^2 of the normal around $\boldsymbol{q}_2$ and σ_ψ^2 around $\boldsymbol{q}_1$.

The point $\mathcal{Z}_0(\boldsymbol{Z}_0)$ closest to the origin is given by

$$\boldsymbol{Z}_0 = D\boldsymbol{N}\,. \tag{19.5}$$

We will represent the coordinates $\boldsymbol{X}_0$ of the centroid $\mathcal{X}_0$ as the sum of two orthogonal vectors $\boldsymbol{Z}_0$ and $\boldsymbol{M}$

$$\boldsymbol{X}_0 = \boldsymbol{Z}_0 + \boldsymbol{M} = Q(D\boldsymbol{N}'' + \boldsymbol{M}'')\,. \tag{19.6}$$

see Fig. 19.2, and – represented in the rotated coordinate system (see Fig. 19.2 right) –

$$\boldsymbol{N}'' = \boldsymbol{N}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{e}_3 \quad \boldsymbol{M}'' = \begin{bmatrix} M_X'' \\ M_Y'' \\ 0 \end{bmatrix}\ . \tag{19.7}$$

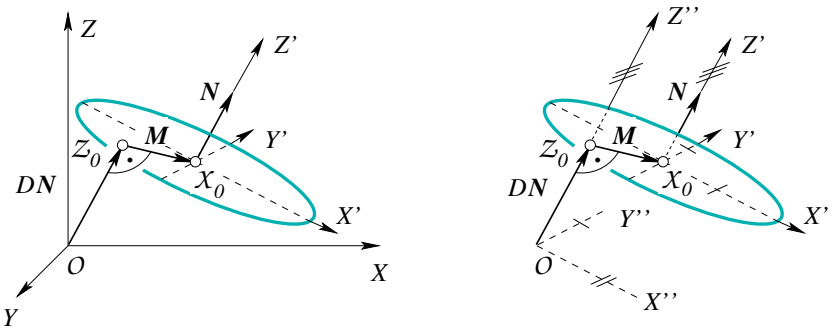


Figure 19.2: Representation of uncertain plane. **Left:** Relation between global frame (XYZ) and the local frame $(X'Y'Z')$. **Right:** Relation between the global system rotated by Q $(X''Y''Z'')$ and the local frame $(X'Y'Z')$, which are parallel

19.2.2 Covariance Matrix of the Plane Parameters

The standard deviations can be derived by transforming the points into the coordinate system $(X'Y'Z')$ of their weighted centroid. Then we only have three uncertain parameters collected in the 3-vector

$$\boldsymbol{A}^\circ = \begin{bmatrix} A_1^\circ \\ A_2^\circ \\ A_3^\circ \end{bmatrix} . \tag{19.8}$$

We have

- the uncertain Z'-coordinate A_1° of the centroid, and
- the uncertain X'- and Y'-coordinates (A_2°, A_3°) of the normal.

Hence we represent the uncertainty of the plane by

$$\mathbb{D}(\underline{\boldsymbol{A}}^\circ) = \begin{bmatrix} \sigma_q^2 & & \\ & \sigma_\phi^2 & \\ & & \sigma_\psi^2 \end{bmatrix} . \tag{19.9}$$

The three parameters are related to the centroid and the normal by

$$\Delta\boldsymbol{A}^* := \begin{bmatrix} \Delta\boldsymbol{X}_0 \\ \Delta\boldsymbol{N} \end{bmatrix} = \begin{bmatrix} Q & \begin{bmatrix} 0 \\ 0 \\ \Delta A_1^\circ \end{bmatrix} \\ Q & \begin{bmatrix} \Delta A_2^\circ \\ \Delta A_3^\circ \\ 0 \end{bmatrix} \end{bmatrix} = J_r(Q)\Delta\boldsymbol{A}^\circ \tag{19.10}$$

with

$$\underset{6\times 3}{J_r(Q)} = \begin{bmatrix} Q & \\ & Q \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{q}_1 & \boldsymbol{q}_2 \end{bmatrix} . \tag{19.11}$$

The covariance matrices of the centroid and the normal then can be given directly. The centroid and the normal are statistically uncorrelated.

The centroid $\underline{\boldsymbol{X}}_0$ is only uncertain across the plane, hence in the direction of the normal

$$\boxed{\Sigma_{X_0X_0} = Q \begin{bmatrix} 0 & & \\ & 0 & \\ & & \sigma_q^2 \end{bmatrix} Q^\mathsf{T} = \sigma_q^2 \boldsymbol{N}\boldsymbol{N}^\mathsf{T} .} \tag{19.12}$$

The uncertainty of the normal $\boldsymbol{N}$ is

$$\boxed{\Sigma_{NN} = Q \begin{bmatrix} \sigma_\phi^2 & & \\ & \sigma_\psi^2 & \\ & & 0 \end{bmatrix} Q^\mathsf{T} = \sigma_\phi^2 \boldsymbol{q}_1 \boldsymbol{q}_1^\mathsf{T} + \sigma_\psi^2 \boldsymbol{q}_2 \boldsymbol{q}_2^\mathsf{T} .} \tag{19.13}$$

Hence the direction of the major uncertainty of the normal of the plane is coded in the covariance matrix. Eqs. (19.12) and (19.13) clarify, why we only need the rotation matrix Q and the three standard deviations σ_q, σ_ϕ, and σ_ψ for representing the uncertainty of the plane. The rotation matrix Q this is responsible for both, the normal and the covariance matrix of the plane.

The inverse relation is

$$\Delta \boldsymbol{A}^\circ = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Q^\mathsf{T} & \\ & Q^\mathsf{T} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{X}_0 \\ \Delta \boldsymbol{N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_3^\mathsf{T} & \boldsymbol{0}^\mathsf{T} \\ \boldsymbol{0}^\mathsf{T} & \boldsymbol{q}_1^\mathsf{T} \\ \boldsymbol{0}^\mathsf{T} & \boldsymbol{q}_2^\mathsf{T} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{X}_0 \\ \Delta \boldsymbol{N} \end{bmatrix} =: J_r^\mathsf{T}(Q) \Delta \boldsymbol{A}^* . \tag{19.14}$$

which has covariance matrix

$$\Sigma_{A^\circ A^\circ} = \begin{bmatrix} \sigma_q^2 & & \\ & \sigma_\phi^2 & \\ & & \sigma_\psi^2 \end{bmatrix} . \tag{19.15}$$

Hence, if a point $\mathcal{X}_i$ lies on the plane $\mathcal{A}$, then the point $\mathcal{X}_i'([X_i', Y_i', 0])$ lies on the plane $\mathcal{A}'$, which is the $X'Y'$-plane. The points $\mathcal{X}_i$ and $\mathcal{X}_i'$ are related by

$$\boldsymbol{X}_i = Q \boldsymbol{X}_i' + \boldsymbol{X}_0 \quad \text{or} \quad \boldsymbol{X}_i' = Q^\mathsf{T} (\boldsymbol{X}_i - \boldsymbol{X}_0) . \tag{19.16}$$

19.3 Uncertain Plane from 3D Points

19.3.1 Fitting a plane through 3D points with isotropic uncertainty

Given are I uncertain 3D points $\mathcal{X}_i, i = 1, ..., I$, with $\{\boldsymbol{X}_i, \sigma_i^2 I_3\}$.

1. We can show that the best fitting plane $\mathcal{A}(\mathbf{A})$ with

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{N} \\ -D \end{bmatrix} \tag{19.17}$$

passes through the weighted centroid $\boldsymbol{X}_0$, that its normal $\boldsymbol{N}$ is the eigenvector of the (unweighted) moment matrix belonging to the smallest eigenvalue, and that it is given by $\boldsymbol{N}^\mathsf{T}(\boldsymbol{X} - \boldsymbol{X}_0) = 0$.
The moment matrix is

$$M = \sum_i w_i (\boldsymbol{X}_i - \boldsymbol{X}_0)(\boldsymbol{X}_i - \boldsymbol{X}_0)^\mathsf{T} = Q \Lambda Q^\mathsf{T} = \lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^\mathsf{T} + \lambda_2 \boldsymbol{q}_2 \boldsymbol{q}_2^\mathsf{T} + \lambda_3 \boldsymbol{q}_3 \boldsymbol{q}_3^\mathsf{T} \tag{19.18}$$

with

$$\boxed{w_i = \frac{1}{\sigma_i^2} \quad \text{and} \quad \boldsymbol{X}_0 = \frac{\sum_i w_i \boldsymbol{X}_i}{\sum w_i}} \tag{19.19}$$

and the rotation matrix,

$$Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3] , \tag{19.20}$$

and the diagonal matrix

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} , \tag{19.21}$$

where the eigenvalues are sorted in decreasing order. The normal is

$$\boxed{\boldsymbol{N} = \boldsymbol{q}_3 = \boldsymbol{e}_3^{\mathsf{T}} Q\,.} \tag{19.22}$$

2. (Exercise) Show the *theoretical variances* of the parameters of a plane through I equally weighted ($w_i = 1$) 3D points $\mathcal{X}_i$ with standard deviation σ for all coordinates can be determined from

$$\sigma_q^2 = \frac{\sigma^2}{I} \qquad \sigma_\phi^2 = \frac{\sigma^2}{\lambda_1} \qquad \sigma_\psi^2 = \frac{\sigma^2}{\lambda_2}\,, \tag{19.23}$$

where σ_q^2 is the variance of the position of the plane in the direction of the normal and σ_ϕ^2 and σ_ψ^2 are the variances of rotations around the two principle axes of the point set.
Hint: Translate the point cloud into the origin and rotate it such that the two major axes of the moment matrix fall into the X- and the Y-coordinate axes. Then apply the reasoning from the chapter on the best fitting 2D line.
Using the weighted moment matrix, for general weights this generalizes to

$$\boxed{\sigma_q^2 = \frac{1}{I\bar{w}} \quad \sigma_\phi^2 = \frac{1}{\lambda_1} \quad \sigma_\psi^2 = \frac{1}{\lambda_2}} \tag{19.24}$$

3. (Exercise) Show that the *estimated variance* of the plane's position q perpendicular to the plane and the two principle normal directions are given by

$$\sigma_q^2 = \frac{1}{I-3}\,\frac{\lambda_3}{I} \qquad \sigma_\phi^2 = \frac{1}{I-3}\,\frac{\lambda_3}{\lambda_1} \qquad \sigma_\psi^2 = \frac{1}{I-3}\,\frac{\lambda_3}{\lambda_2}\,. \tag{19.25}$$

Using the weighted moment matrix, for general weights this generalizes to

$$\boxed{\hat{\sigma}_q^2 = \frac{1}{I-3}\frac{\lambda_3}{I\bar{w}} \quad \hat{\sigma}_\phi^2 = \frac{1}{I-3}\frac{\lambda_3}{\lambda_1} \quad \hat{\sigma}_\psi^2 = \frac{1}{I-3}\frac{\lambda_3}{\lambda_2}\,.} \tag{19.26}$$

19.3.1.1 Relation to moments and recursive estimation

Now we observe, that the parameters specifying an uncertain plane can be uniquely derived from the non-central moments. They allow a simple and possibly recursive estimation of the mean of several planes.

The non-central moments are

$$m_{kln} = \sum_i w_i X_i^k Y_i^l Z_i^n \quad \text{with} \quad k+l+n \in \{0,1,2\} \tag{19.27}$$

namely

$$m_{000} = \sum_i w_i \tag{19.28}$$

$$m_{100} = \sum_i w_i X_i \tag{19.29}$$

$$m_{010} = \sum_i w_i Y_i \tag{19.30}$$

$$m_{001} = \sum_i w_i Z_i \tag{19.31}$$

$$m_{200} = \sum_i w_i X_i^2 \tag{19.32}$$

$$m_{110} = \sum_i w_i X_i Y_i \tag{19.33}$$

$$m_{101} = \sum_i w_i X_i Z_i \tag{19.34}$$

$$m_{020} = \sum_i w_i Y_i^2 \tag{19.35}$$

$$m_{011} = \sum_i w_i Y_i Z_i \tag{19.36}$$

$$m_{002} = \sum_i w_i Z_i^2 \tag{19.37}$$

together with the number of points

$$I = \sum_i 1 \,. \tag{19.38}$$

Especially we have

$$I\overline{w} = m_{000} \tag{19.39}$$

$$X_0 = m_{100}/m_{000} \tag{19.40}$$

$$Y_0 = m_{010}/m_{000} \tag{19.41}$$

$$Z_0 = m_{001}/m_{000} \tag{19.42}$$

$$\mu_{200} = m_{200}/m_{000} - X_0^2 \tag{19.43}$$

$$\mu_{110} = m_{110}/m_{000} - X_0 Y_0 \tag{19.44}$$

$$\mu_{101} = m_{101}/m_{000} - X_0 Z_0 \tag{19.45}$$

$$\mu_{020} = m_{020}/m_{000} - Y_0^2 \tag{19.46}$$

$$\mu_{011} = m_{011}/m_{000} - Y_0 Z_0 \tag{19.47}$$

$$\mu_{002} = m_{002}/m_{000} - Z_0^2 \tag{19.48}$$

$$M = \begin{bmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{bmatrix} . \tag{19.49}$$

The eigenvalues of the moment matrix yield the variances of the position and the normal via (19.26). Hence we have a mapping from the moments $\boldsymbol{m}$ (including the number of points I) to the centroid form $\boldsymbol{c}$ of the plane

$$\boxed{\boldsymbol{m} \mapsto \boldsymbol{c}: \qquad \boldsymbol{c} = \boldsymbol{c}(\boldsymbol{m}) \quad \text{or} \quad \left\{ \boldsymbol{X}_0, Q; \sigma_q^2, \sigma_\phi^2, \sigma_\psi^2 \right\} \leftarrow \{m_{000}, ..., m_{002}, I\} \,.} \tag{19.50}$$

19.3.2 Fitting a plane through a set of 3D points with arbitrary covariance matrix

We can assume to have approximate values, thus only need to update these using an iterative scheme, where often only one iteration is necessary.

19.3.2.1 An iterative solution

We start from the nonlinear constraints

$$g_i(\widehat{\boldsymbol{X}}_i; \widehat{\boldsymbol{N}}, \hat{D}) = \widehat{\boldsymbol{N}}^{\mathsf{T}} \widehat{\boldsymbol{X}}_i - \hat{D} = 0 \tag{19.51}$$

where D is the distance of the plane to the origin. We will later find the centroid $\boldsymbol{X}_0$ on the plane. In addition, we have the length constraint for the normal

$$h(\widehat{\boldsymbol{N}}) = \frac{1}{2}\left(|\widehat{\boldsymbol{N}}|^2 - 1\right) = 0 \tag{19.52}$$

Starting from approximate values for the unknown parameters and the fitted observations we thus have the linearized model

$$g_i(\widehat{\boldsymbol{X}}_i; \widehat{\boldsymbol{N}}, \hat{D}) = \widehat{\boldsymbol{N}}^{a\mathsf{T}} \widehat{\boldsymbol{X}}_i^a - \hat{D}^a + \widehat{\boldsymbol{N}}^{a\mathsf{T}} \widehat{\Delta \boldsymbol{X}}_i + \widehat{\boldsymbol{X}}_i^{a\mathsf{T}} \widehat{\Delta \boldsymbol{N}} + \widehat{\Delta D} = 0 \tag{19.53}$$

or

$$g_i(\widehat{\boldsymbol{X}}_i; \widehat{\boldsymbol{N}}, \hat{D}) = g_i(\widehat{\boldsymbol{X}}_i^a; \widehat{\boldsymbol{N}}^a, \hat{D}^a) + \boldsymbol{a}_i^{\mathsf{T}} \widehat{\Delta \boldsymbol{\theta}} + \boldsymbol{b}_i^{\mathsf{T}} \widehat{\Delta \boldsymbol{y}} = 0 \tag{19.54}$$

with the corrections to the unknown parameters, collected in a 4-vector

$$\widehat{\Delta \boldsymbol{\theta}} := \begin{bmatrix} \widehat{\Delta \boldsymbol{N}} \\ \widehat{\Delta D} \end{bmatrix}, \quad \widehat{\Delta \boldsymbol{y}_i} := \widehat{\Delta \boldsymbol{X}}_i, \quad \boldsymbol{a}_i := \begin{bmatrix} \widehat{\boldsymbol{X}}_i^a \\ 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{b}_i := \boldsymbol{N}^a. \tag{19.55}$$

Therefore we have the normal equations

$$M \widehat{\Delta \boldsymbol{p}} = \boldsymbol{m} \quad \text{or} \quad \begin{bmatrix} N & H \\ H^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta \boldsymbol{\theta}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} \\ c_h \end{bmatrix} \tag{19.56}$$

with

$$N = \sum_i w_{q_i} \boldsymbol{a}_i \boldsymbol{a}_i^{\mathsf{T}}, \tag{19.57}$$

$$\boldsymbol{n} = \sum_i w_{q_i} (\boldsymbol{a}_i (-g_i + \boldsymbol{b}_i^{\mathsf{T}} (\widehat{\boldsymbol{X}}_i - \boldsymbol{X}_i)), \tag{19.58}$$

$$H = \widehat{\boldsymbol{N}}^a, \tag{19.59}$$

$$c_g = -(|\widehat{\boldsymbol{N}}^a|^2 - 1) \tag{19.60}$$

$$w_{q_i} = \frac{1}{\boldsymbol{b}_i^{\mathsf{T}} \Sigma_{y_i y_i} \boldsymbol{b}_i} := \frac{1}{\boldsymbol{N}^{a\mathsf{T}} \Sigma_{X_i X_i} \boldsymbol{N}^a} \tag{19.61}$$

We use the following update for the normal

$$\widehat{\boldsymbol{N}} = \mathsf{N}(\widehat{\boldsymbol{N}}^a + \Delta \widehat{\boldsymbol{N}}) \tag{19.62}$$

The covariance matrix of the parameters results from the inverse of the normal equation matrix, or, when eliminating the Lagrangian parameter from the 4×4 matrix

$$\Sigma_{\hat{\theta}\hat{\theta}} = (N + HH^{\mathsf{T}})^{-1} - HH^{\mathsf{T}} = \begin{bmatrix} \Sigma_{\widehat{N}\widehat{N}} & \Sigma_{\widehat{N}\hat{D}} \\ \Sigma_{\hat{D}\widehat{N}} & \Sigma_{\hat{D}\hat{D}} \end{bmatrix}, \tag{19.63}$$

which has rank 3, and generally is a full matrix.

19.3.2.2 Choosing the Local Coordinate System

We now choose the points reduced to some reference frame with center $\boldsymbol{X}_0$ and axes Q

$$\boldsymbol{X}_i = Q \boldsymbol{X}_i' + \boldsymbol{X}_0 \quad \text{or} \quad \boldsymbol{X}_i' = Q^{\mathsf{T}} (\boldsymbol{X}_i - \boldsymbol{X}_0), \tag{19.64}$$

In homogeneous coordinates this is

$$\begin{bmatrix} \boldsymbol{X}_i \\ 1 \end{bmatrix} = \begin{bmatrix} Q & \boldsymbol{X}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}'_i \\ 1 \end{bmatrix} \tag{19.65}$$

The plane therefore transforms as

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{A}_h \\ A_0 \end{bmatrix} = \begin{bmatrix} Q & \boldsymbol{X}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}^\mathsf{-} = \begin{bmatrix} \boldsymbol{A}'_h \\ A'_0 \end{bmatrix} = \mathbf{A}' \tag{19.66}$$

or explicitly

$$\begin{bmatrix} \boldsymbol{A}_h \\ A_0 \end{bmatrix} = \begin{bmatrix} Q & \mathbf{0} \\ -\boldsymbol{X}_0^\mathsf{T} Q & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}'_h \\ A'_0 \end{bmatrix}. \tag{19.67}$$

The normal $\boldsymbol{N} = \boldsymbol{A}_h$ therefore is transformed as

$$\boldsymbol{N} = Q\boldsymbol{N}' \quad \text{or} \quad \boldsymbol{N}' = Q^\mathsf{T}\boldsymbol{N}. \tag{19.68}$$

The distance $D = -A_0$ to the origin is transferred as

$$D = D' + \boldsymbol{X}_0^\mathsf{T} Q\boldsymbol{N}' = D' + \boldsymbol{X}_0^\mathsf{T}\boldsymbol{N} \quad \text{or} \quad D' = D - \boldsymbol{N}^\mathsf{T}\boldsymbol{X}_0. \tag{19.69}$$

The covariance matrices transform as

$$\begin{aligned} \mathbb{D}(\mathbf{A}) &= \begin{bmatrix} \Sigma_{NN} & \Sigma_{ND} \\ \Sigma_{DN} & \sigma_D^2 \end{bmatrix} && (19.70)\\ &= \begin{bmatrix} Q & 0 \\ \boldsymbol{X}_0^\mathsf{T} Q & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{N'N'} & \mathbf{0} \\ \mathbf{0} & \sigma_{D'}^2 \end{bmatrix} \begin{bmatrix} Q^\mathsf{T} & Q^\mathsf{T}\boldsymbol{X}_0 \\ 0 & 1 \end{bmatrix} && (19.71)\\ &= \begin{bmatrix} Q\Sigma_{N'N'}Q^\mathsf{T} & Q\Sigma_{N'N'}Q^\mathsf{T}\boldsymbol{X}_0 \\ \boldsymbol{X}_0^\mathsf{T} Q\Sigma_{N'N'}Q^\mathsf{T} & \boldsymbol{X}_0^\mathsf{T} Q\Sigma_{N'N'}Q^\mathsf{T}\boldsymbol{X}_0 + \sigma_{D'}^2 \end{bmatrix} && (19.72) \end{aligned}$$

and

$$\begin{aligned} \mathbb{D}(\mathbf{A}') &= \begin{bmatrix} \Sigma_{N'N'} & \Sigma_{N'D'} \\ \Sigma_{D'N'} & \sigma_{D'}^2 \end{bmatrix} && (19.73)\\ &= \begin{bmatrix} Q^\mathsf{T} & 0 \\ -\boldsymbol{X}_0^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{NN} & \mathbf{0} \\ \mathbf{0} & \sigma_D^2 \end{bmatrix} \begin{bmatrix} Q^\mathsf{T} & -\boldsymbol{X}_0 \\ 0 & 1 \end{bmatrix} && (19.74)\\ &= \begin{bmatrix} Q^\mathsf{T}\Sigma_{NN}Q & -Q^\mathsf{T}\Sigma_{NN}\boldsymbol{X}_0 \\ -\boldsymbol{X}_0^\mathsf{T}\Sigma_{NN}Q & \boldsymbol{X}_0^\mathsf{T}\Sigma_{NN}\boldsymbol{X}_0 + \sigma_D^2 \end{bmatrix} && (19.75) \end{aligned}$$

We first choose Q such that the covariance matrix

$$\Sigma_{\widehat{N}\widehat{N}} = Q\Sigma_{\widehat{N}'\widehat{N}'}Q^\mathsf{T} = Q \begin{bmatrix} \sigma_{\hat{\phi}}^2 & & \\ & \sigma_{\hat{\psi}}^2 & \\ & & 0 \end{bmatrix} Q^\mathsf{T} \tag{19.76}$$

of the normal is diagonal, which can be achieved by an eigenvalue decomposition of $\Sigma_{\widehat{N}\widehat{N}}$. Then the normal is (19.4)

$$\underline{\widehat{\boldsymbol{N}}}' = N\left(\begin{bmatrix} \underline{\hat{N}}'_X \\ \underline{\hat{N}}'_Y \\ 1 \end{bmatrix}\right) = N\left(\begin{bmatrix} \underline{\widehat{\boldsymbol{N}}}'_r \\ 1 \end{bmatrix}\right) = Q^\mathsf{T}\underline{\boldsymbol{N}}. \tag{19.77}$$

After the diagonalization we obtain the covariance matrix, where the distance D'' is in the rotated and not yet translated system:

$$\mathbb{D}\left(\begin{bmatrix} \underline{\widehat{\boldsymbol{N}}}' \\ \underline{\hat{D}}'' \end{bmatrix}\right) = \begin{bmatrix} \mathrm{Diag}([\sigma_{\hat{\phi}}^2, \sigma_{\hat{\psi}}^2]) & \mathbf{0} & \Sigma_{\widehat{N}'_r\hat{D}''} \\ \mathbf{0}^\mathsf{T} & 0 & 0 \\ \Sigma_{\hat{D}''\widehat{N}'_r} & 0 & \sigma_{\hat{D}''}^2 \end{bmatrix} \tag{19.78}$$

with

$$\mathbb{D}\left(\begin{bmatrix} \underline{N}'_X \\ \underline{N}'_Y \end{bmatrix}\right) = \mathrm{Diag}([\sigma^2_{\hat{\phi}}, \sigma^2_{\hat{\psi}}]) = \begin{bmatrix} \boldsymbol{q}_1^\mathsf{T} \\ \boldsymbol{q}_2^\mathsf{T} \end{bmatrix} \Sigma_{\widehat{N}\widehat{N}} \, [\boldsymbol{q}_1, \boldsymbol{q}_2] \tag{19.79}$$

and

$$\mathbb{C}\mathrm{ov}\left(\begin{bmatrix} \underline{N}'_X \\ \underline{N}'_Y \end{bmatrix}, \hat{D}''\right) = \begin{bmatrix} \sigma_{\widehat{N}'_X \hat{D}''} \\ \sigma_{\widehat{N}'_Y \hat{D}''} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1^\mathsf{T} \\ \boldsymbol{q}_2^\mathsf{T} \end{bmatrix} \Sigma_{\widehat{N}\hat{D}} \, . \tag{19.80}$$

Next we choose $\boldsymbol{X}_0$ such that the uncertainty of a point across the plane is minimum. An arbitrary point $\mathcal{X}(\boldsymbol{X})$ has the distance

$$\underline{D}_X = \underline{\boldsymbol{N}}^\mathsf{T} \boldsymbol{X} - \underline{D} = [\boldsymbol{X}^\mathsf{T}, -1] \begin{bmatrix} \underline{\widehat{\boldsymbol{N}}} \\ -\underline{\hat{D}} \end{bmatrix} . \tag{19.81}$$

Its variance is

$$\sigma^2_{D_X} = \boldsymbol{X}^\mathsf{T} \Sigma_{\widehat{N}\widehat{N}} \boldsymbol{X} - 2\Sigma_{\hat{D}\widehat{N}} \boldsymbol{X} + \sigma^2_{\hat{D}} \, . \tag{19.82}$$

From its derivative w.r.t. dX

$$\frac{\partial \sigma^2_{D_X}}{\partial \boldsymbol{X}} = 2\Sigma_{\widehat{N}\widehat{N}} \boldsymbol{X} - 2\Sigma_{\hat{D}\widehat{N}} \tag{19.83}$$

In the rotated system we have

$$\frac{\partial \sigma^2_{D''_X}}{\partial \boldsymbol{X}''} = 2\Sigma_{\widehat{N}'\widehat{N}'} \boldsymbol{X}'' - 2\Sigma_{\hat{D}''\widehat{N}''} = 2 \begin{bmatrix} \sigma^2_{\hat{\phi}} & 0 & 0 \\ 0 & \sigma^2_{\hat{\psi}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M''_X \\ M''_X \\ 0 \end{bmatrix} . \tag{19.84}$$

from which we obtain

$$\begin{bmatrix} M''_X \\ M''_Y \end{bmatrix} = - \begin{bmatrix} \sigma^2_{\hat{\phi}} & 0 \\ 0 & \sigma^2_{\hat{\psi}} \end{bmatrix}^{-1} \begin{bmatrix} \sigma_{\widehat{N}'_X \hat{D}''} \\ \sigma_{\widehat{N}'_Y \hat{D}''} \end{bmatrix} = - \begin{bmatrix} \sigma^2_{\hat{\phi}} & 0 \\ 0 & \sigma^2_{\hat{\psi}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{q}_1^\mathsf{T} \\ \boldsymbol{q}_2^\mathsf{T} \end{bmatrix} \Sigma_{\widehat{N}\hat{D}} \tag{19.85}$$

Finally, we have the centroid

$$\boldsymbol{X}_0 = \widehat{\boldsymbol{N}} - \mathcal{Q} \begin{bmatrix} \sigma^2_{\hat{\phi}} & 0 & 0 \\ 0 & \sigma^2_{\hat{\psi}} & 0 \\ 0 & 0 & 0 \end{bmatrix}^+ \mathcal{Q}^\mathsf{T} \Sigma_{\widehat{N}\hat{D}} \, . \tag{19.86}$$

or

$$\boxed{\boldsymbol{X}_0 = \widehat{\boldsymbol{N}} - \Sigma^+_{\widehat{N}\hat{D}} \Sigma_{\widehat{N}\hat{D}} \, .} \tag{19.87}$$

Remark: This is in full analogy to the centroid of the 2D line when using the covariance matrix of the normal

$$\Sigma_{\hat{n}\hat{n}} = \sigma^2_{\hat{\phi}} \boldsymbol{n}^\perp \boldsymbol{n}^{\perp\mathsf{T}} \quad \text{and} \quad \Sigma^+_{\hat{n}\hat{n}} = \sigma^{-2}_{\hat{\phi}} \boldsymbol{n}^\perp \boldsymbol{n}^{\perp\mathsf{T}} \quad \text{with} \quad \boldsymbol{n} = \begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix} \quad \text{and} \quad \boldsymbol{n}^\perp = \begin{bmatrix} \cos\alpha \\ -\sin\alpha \end{bmatrix} \tag{19.88}$$

and the covariance with the distance

$$\Sigma_{\hat{n}\hat{d}} = \sigma_{\phi d} \boldsymbol{n}^\perp \tag{19.89}$$

since

$$\begin{aligned} \boldsymbol{x}_0 &= \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} m_0 \\ d \end{bmatrix} && (19.90) \\ &= \begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix} d + \begin{bmatrix} \cos\alpha \\ -\sin\alpha \end{bmatrix} m_0 && (19.91) \\ &= \boldsymbol{n} - \begin{bmatrix} \cos\alpha \\ -\sin\alpha \end{bmatrix} \left(-\sigma^{-2}_{\phi} \sigma_{\phi d}\right) && (19.92) \\ &= \boldsymbol{n} - \Sigma^+_{\hat{n}\hat{n}} \Sigma_{\hat{n}\hat{d}} \, . && (19.93) \end{aligned}$$

$\diamond$

19.3.3 Checking a Set of Points for Planarity

19.3.3.1 Assuming the uncertainty of the points is known

We test whether the surface consisting of I points is planar, testing the null hypothesis

$$H_{01}: \quad \hat{\sigma}_0^2 = 1 \tag{19.94}$$

versus the alternative hypothesis

$$H_{a1}: \quad \hat{\sigma}_0^2 > 1 \tag{19.95}$$

using the chi-square test statistic

$$\underline{X}|H_{01} = \underline{\Omega}|H_{01} \sim \chi_R^2 \,. \tag{19.96}$$

which is χ_R^2-distributed under the null hypothesis. If the test is rejected, this may be caused

- by a too small standard deviation of the given points, or
- by a significant deviation of the surface from a plane, or
- both.

Remark: The degrees of freedom R should not be taken too large, since otherwise the null-hypothesis always will be rejected, see the discussion in Förstner and Wrobel (2016), around Eq. (4.88). ◇

19.3.3.2 Assuming an estimate of the uncertainty of the points of the plane is not known

We assume, the variance factor $\hat{\sigma}_{0a}^2$ of all given points may be taken from a robust estimate of all variance factors. Its degrees of freedom is assumed to be R_0.

We test the null hypothesis for the current plane

$$H_{02}: \quad \hat{\sigma}_0^2 = \hat{\sigma}_{0a}^2 \tag{19.97}$$

against the alternative hypothesis

$$H_{a2}: \quad \hat{\sigma}_0^2 > \hat{\sigma}_{0a}^2 \tag{19.98}$$

using the Fisher test statistic

$$\underline{F}|H_{02} = \frac{\underline{\hat{\sigma}_0^2}}{\underline{\hat{\sigma}_{0a}^2}}|H_{02} \sim F_{R,R_0} \tag{19.99}$$

which is F_{R,R_0}-distributed under the null hypothesis.

19.4 Estimating a Mean Plane

Given are I planes $\mathcal{A}_i$, the task is to find the best estimate for the mean plane $\mathbf{A}$.

We discuss three solutions:

1. A solution based on moments of the point cloud, assuming isotropic uncertainty.
2. A statistically suboptimal solution for the.
3. A statistically optimal solution based on the centroid representation.

19.4.1 Estimating the mean plane using moments

Let us assume we have J patches, represented by their moment vector $\boldsymbol{m}_j$.

Obviously, it is simple to derive the mean plane. We just need to add all non-central moments. Hence:

$$\boxed{\boldsymbol{m} = \sum_{j=1}^{J} \boldsymbol{m}_j\,.} \tag{19.100}$$

The parameters of the uncertain mean plane can then be derived from $\boldsymbol{c}(\boldsymbol{m})$, see (19.50). We need to observe:

1. Eq. (19.100) allows a recursive estimation of the plane. Let the mean plane derived from the first j patches be

   $$\boldsymbol{m}^{(j)} = \sum_{k=1}^{j} \boldsymbol{m}_k\,. \tag{19.101}$$

   Then adding the $(j+1)$-th patch leads to

   $$\boxed{\boldsymbol{m}^{(j+1)} = \boldsymbol{m}^{(j)} + \boldsymbol{m}_{j+1}\,.} \tag{19.102}$$

2. In a similar manner a patch k can be deleted if $k \in \{1, ..., j\}$:

   $$\boldsymbol{m}^{(j\backslash k)} = \boldsymbol{m}^{(j)} - \boldsymbol{m}_k\,. \tag{19.103}$$

3. Before inserting a patch into the list of patches, a statistical test could be performed. This can be based on the difference vector of the new patch $\mathcal{A}_{j+1}$ and the current mean plane $\mathcal{A}^{(j)}$

   $$\boldsymbol{d} = J_r^{\mathsf{T}}(\boldsymbol{\mu}_A)(\boldsymbol{A}_{j+1} - \boldsymbol{A}^{(j)}) \tag{19.104}$$

   and its covariance matrix

   $$\Sigma_{dd} = J_r^{\mathsf{T}}(\boldsymbol{\mu}_A)\left(\Sigma_{A_{j+1}A_{j+1}} + \Sigma_{A^{(j)}A^{(j)}}\right) J_r(\boldsymbol{\mu}_A) \tag{19.105}$$

   leading to the test statistic

   $$\underline{T} = \underline{\boldsymbol{d}}^{\mathsf{T}}\Sigma_{dd}^{-1}\underline{\boldsymbol{d}} \sim \chi^2_3\,. \tag{19.106}$$

   Observe, the vector $\boldsymbol{d}$ in (19.104) is the difference $\boldsymbol{A}_{j+1,r} - \boldsymbol{A}_r^{(j)}$ of the reduced plane coordinates assuming the common tangent plane is given by $\boldsymbol{\mu}_A$. The argument $\boldsymbol{\mu}_A$ of $J(\boldsymbol{\mu}_A)$ best is chosen as the current mean plane $\boldsymbol{\mu}_A := \mathbf{A}^{(j)}$.

   Here we assume, the planes are Euclideanly normalized, see (19.17), i.e., the normal has length 1. Then the projection matrix $J_r(\mathbf{A})$ is given by

   $$\boxed{\underset{4\times 3}{J_r(\mathbf{A})} = \begin{bmatrix} \underset{3\times 2}{J_r(\boldsymbol{N})} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \quad \text{with} \quad J_r(\boldsymbol{N}) = \mathsf{null}(\boldsymbol{N}^{\mathsf{T}})\,.} \tag{19.107}$$

4. All moments need to refer to the same coordinate system. Therefore, it might be useful to condition all coordinates before determining and fusing all patches.
5. There is no non-linearity involved in the recursive *estimation* involved, if we only consider the moments. The non-linearity only refers to deriving the centroid or other parameters of the planes. Especially no directions or angles are involved. A recursive determination of the variances would be difficult, without going back to the moments.
6. Eq. (19.102) can also be specialized to including a single point.
7. The whole procedure could once be repeated with modified weights. If the weights are reduced to 0, this is equivalent to deleting previously included patches, which can be done using (19.103).

Hence, the moments, the 11 parameters including the number of points, can be interpreted as the memory generating the current version of the plane. In statistical terms, the moments are sufficient test statistics, i.e., no other information is necessary to perform the estimation.

19.4.2 Approximate estimating the mean plane using plane parameters

Let us assume we have J patches, represented by their homogeneous vector $\mathbf{A}_j = [\boldsymbol{N}_j^\mathsf{T}, -D_j]^\mathsf{T}$ together with the covariance matrix of the reduced vector, namely $\left\{\left(A_j, \Sigma_{A_{jr}A_{jr}}\right)\right\}$.

The constraint, that the individual patch is identical to the mean plane is given by

$$\overline{\mathsf{M}}(\mathbf{A}_j)\mathbf{A} = -\overline{\mathsf{M}}(\mathbf{A})\mathbf{A}_j = \mathbf{0} \quad \text{with} \quad \overline{\mathsf{M}}(\mathbf{A}) = \begin{bmatrix} 0 & -N_Z & N_Y & 0 \\ N_Z & 0 & -N_X & 0 \\ -N_Y & N_X & 0 & 0 \\ -D & 0 & 0 & -N_X \\ 0 & -D & 0 & -N_Y \\ 0 & 0 & -D & -N_Z \end{bmatrix}, \tag{19.108}$$

see Förstner and Wrobel (2016, Eq. (7.41)). Since a plane has only three degrees of freedom, we need to select three constraints from the six constraints in (19.108). If the coordinate system is chosen such that all distances D_j are non-zero, the last three constraints may be used. leading to

$$C_j^\mathsf{T}\overline{\mathsf{M}}(\mathbf{A}_j)\mathbf{A} = -C_j^\mathsf{T}\overline{\mathsf{M}}(\mathbf{A})\mathbf{A}_j = \mathbf{0} \quad \text{with} \quad C_j^\mathsf{T} = [0_3 \mid I_3]\,, \tag{19.109}$$

or

$$\boldsymbol{g}(\mathbf{A}_j, \mathbf{A}) = X_j\mathbf{A} = Z_j\mathbf{A}_j = \mathbf{0} \quad \text{with} \quad X_j^\mathsf{T} = [-D_j I_3 \mid -\boldsymbol{N}_j] \quad \text{and} \quad Z_j = [DI_3 \mid \boldsymbol{N}]\,. \tag{19.110}$$

Observe, the Jacobians of the constraint $\boldsymbol{g}(\boldsymbol{A}_j, \boldsymbol{A})$ w.r.t. the unknown parameters and the observations are X_j and Z_j, the last matrix is not depending on j.

For estimating the plane, we concatenate all $3J$ constraints in the following form

$$\boldsymbol{g}(\{\mathbf{A}_j\}, \mathbf{A}) = \underset{3J\times 4}{X}\,\mathbf{A} = \mathbf{0}\,. \tag{19.111}$$

The right singular vector of $\mathbf{X}$ belonging to the smallest singular value is the algebraically optimal mean plane, and can be determined using the SVD of X:

$$\boxed{\widehat{\mathbf{A}} = V_{:,4} \quad \text{with} \quad X = USV^\mathsf{T}\,.} \tag{19.112}$$

For deriving the covariance matrix of this solution, we start with the differential of $\boldsymbol{g}$:

$$\mathrm{d}\boldsymbol{g}(\boldsymbol{y}, \mathbf{A}) = \underbrace{XJ_r(\mathbf{A})}_{3J\times 3}\,\underset{3\times 1}{\mathrm{d}\boldsymbol{A}_r} + \underbrace{Z}_{3J\times 3J}\,\underset{3J\times 1}{\mathrm{d}\boldsymbol{y}} \tag{19.113}$$

with

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{A}_{1r} \\ \ldots \\ \boldsymbol{A}_{jr} \\ \ldots \\ \boldsymbol{A}_{Jr} \end{bmatrix}, \quad \boldsymbol{A}_{jr} = J_r^\mathsf{T}(\mathbf{A})\,\mathbf{A}_j \quad \text{and} \quad Z = \mathrm{Diag}(\{\underset{4\times 3}{Z_j}\,\underset{3\times 4}{J_r(\mathbf{A})}\})\,. \tag{19.114}$$

With the reduced coefficient matrix

$$X_r = XJ_r(\mathbf{A}) \tag{19.115}$$

we thus obtain the differential estimates

$$\widehat{\mathrm{d}\boldsymbol{A}}_r = -X_r^+\,Z\mathrm{d}\boldsymbol{y} = -(X_r^\mathsf{T}X_r)^{-1}X_r^\mathsf{T}Z\,\mathrm{d}\boldsymbol{y}\,. \tag{19.116}$$

Hence, we have the covariance matrix of the estimated reduced plane parameters

$$\boxed{\Sigma_{\hat{A}_r\hat{A}_r} = X_r^+\,Z\,\Sigma_{yy}\,Z^\mathsf{T}\,X_r^{+\mathsf{T}}\,.} \tag{19.117}$$

where

$$\Sigma_{yy} = \text{Diag}\left(\{\Sigma_{y_j y_j}\}\right) = \text{Diag}\left(\left\{J_r^{\mathsf{T}}(\mathbf{A})\Sigma_{A_j A_j} J_r(\mathbf{A})\right\}\right) . \tag{19.118}$$

Observe, the solution is suboptimal, since the pseudo inverse X_r^+ is taken instead of the weighted pseudo inverse $(X_{r,\mathsf{W}_{ll}})^+ = (X_r^{\mathsf{T}} W_{yy} X_r)^{-1} X_r^{\mathsf{T}} W_{yy}$. Finally, we obtain the covariance matrix of the estimated mean plane

$$\Sigma_{\hat{A}\hat{A}} = J_r(\mathbf{A})\, \Sigma_{\hat{A}_r\hat{A}_r}\, J_r^{\mathsf{T}}(\mathbf{A}) , \tag{19.119}$$

which has rank 3.

19.4.3 An optimal solution based on the centroid representation

We assume we have given the planes in centroid form,

$$\mathcal{A}_i : \quad \{\boldsymbol{X}_{0i}, Q_i; \sigma_{q_i}, \sigma_{\phi_i}, \sigma_{\psi_i}\} , \tag{19.120}$$

and want to determine the mean plane, also in centroid form

$$\mathcal{A} : \quad \{\boldsymbol{X}_0, Q; \sigma_q, \sigma_\phi, \sigma_\psi\} . \tag{19.121}$$

We use the following nonlinear constraints:

$$\widehat{\boldsymbol{N}} \times \widehat{\boldsymbol{N}}_i = \mathbf{0} , \tag{19.122}$$

which represents two degrees of freedom. We select two independent constraints:

$$M_i^{(s)} S(\widehat{\boldsymbol{N}}) \widehat{\boldsymbol{N}}_i = \mathbf{0} \tag{19.123}$$

and the translational constraint

$$\widehat{\boldsymbol{N}}^{\mathsf{T}} (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0) = 0 , \tag{19.124}$$

which represents the third degree of freedom. For proofs we will use

$$M_i^{(s)} = M^{(s)}(\boldsymbol{N}_i) = M^{(s)}(\boldsymbol{q}_{i3}) = \begin{bmatrix} \boldsymbol{q}_{i1}^{\mathsf{T}} \\ \boldsymbol{q}_{i2}^{\mathsf{T}} \end{bmatrix} \quad \text{with} \quad M_i^{(s)} M_i^{(s)\mathsf{T}} = I_2 , \tag{19.125}$$

hence

$$M_i^{(s)} S(\widehat{\boldsymbol{N}}_i) = \begin{bmatrix} \boldsymbol{q}_{i1}^{\mathsf{T}} \\ \boldsymbol{q}_{i2}^{\mathsf{T}} \end{bmatrix} S(\boldsymbol{q}_3) = \begin{bmatrix} \boldsymbol{q}_2^{\mathsf{T}} \\ -\boldsymbol{q}_1^{\mathsf{T}} \end{bmatrix} \tag{19.126}$$

Hence the nonlinear constraints are

$$\boldsymbol{g}_i(\widehat{\boldsymbol{N}}_i, \widehat{\boldsymbol{X}}_{0i}; \widehat{\boldsymbol{N}}, \widehat{\boldsymbol{X}}_0) = \begin{bmatrix} \widehat{\boldsymbol{N}}^{\mathsf{T}} (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0) \\ M_i^{(s)} S(\widehat{\boldsymbol{N}}) \widehat{\boldsymbol{N}}_i \end{bmatrix} = \mathbf{0} , \quad i = 1, \ldots, I . \tag{19.127}$$

19.4.3.1 The Iterative Solution

We also can assume approximate values, thus can update them using an iterative scheme. Linearization of the constraints yields

$$\begin{aligned} \boldsymbol{g}_i(\widehat{\boldsymbol{N}}_i, \widehat{\boldsymbol{X}}_{0i}; \widehat{\boldsymbol{N}}, \widehat{\boldsymbol{X}}_0) &= \boldsymbol{g}_i(\widehat{\boldsymbol{N}}_i^a, \widehat{\boldsymbol{X}}_{0i}^a; \widehat{\boldsymbol{N}}^a, \widehat{\boldsymbol{X}}_0^a) && (19.128) \\ &+ \begin{bmatrix} (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0)^{\mathsf{T}} \widehat{\Delta \boldsymbol{N}} + \widehat{\boldsymbol{N}}^{\mathsf{T}} \widehat{\Delta \boldsymbol{X}_{0i}} - \widehat{\boldsymbol{N}}^{\mathsf{T}} \widehat{\Delta \boldsymbol{X}_0} \\ -M^{(s)} S(\widehat{\boldsymbol{N}}_i) \widehat{\Delta \boldsymbol{N}} + M^{(s)} S(\widehat{\boldsymbol{N}}) \widehat{\Delta \boldsymbol{N}_i} \end{bmatrix} && (19.129) \end{aligned}$$

This can be written as

$$\boldsymbol{g}_i(\widehat{\boldsymbol{N}}_i, \widehat{\boldsymbol{X}}_{0i}; \widehat{\boldsymbol{N}}, \widehat{\boldsymbol{X}}_0) = \boldsymbol{g}_i(\widehat{\boldsymbol{N}}_i^a, \widehat{\boldsymbol{X}}_{0i}^a; \widehat{\boldsymbol{N}}^a, \widehat{\boldsymbol{X}}_0^a) + X_i \widehat{\Delta \boldsymbol{\theta}} + Z_i^{\mathsf{T}} \widehat{\Delta \boldsymbol{y}_i} \tag{19.130}$$

with

$$\widehat{\Delta\boldsymbol{\theta}} = J_r^{\mathsf{T}} \begin{bmatrix} \Delta\boldsymbol{X}_0 \\ \Delta\boldsymbol{N} \end{bmatrix} = \widehat{\Delta\boldsymbol{A}}^{\circ} \quad \text{and} \quad \widehat{\Delta\boldsymbol{y}}_i = J_r^{\mathsf{T}} \begin{bmatrix} \Delta\boldsymbol{X}_{0i} \\ \Delta\boldsymbol{N}_i \end{bmatrix} = \widehat{\Delta\boldsymbol{A}}_i^{\circ} \tag{19.131}$$

and, since $\widehat{\boldsymbol{N}}_i \equiv \widehat{\boldsymbol{N}}$,

$$X_i = \begin{bmatrix} -\widehat{\boldsymbol{N}}^{\mathsf{T}} & (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0)^{\mathsf{T}} \\ 0_{2\times 3} & -M^{(s)} S(\widehat{\boldsymbol{N}}) \end{bmatrix} J_r \quad \text{and} \quad Z_i^{\mathsf{T}} = \begin{bmatrix} \widehat{\boldsymbol{N}}^{\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} \\ 0_{2\times 3} & M^{(s)} S(\widehat{\boldsymbol{N}}) \end{bmatrix} J_r = \begin{bmatrix} 1 & \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{0} & I_2 \end{bmatrix} = I_3 \,. \tag{19.132}$$

The weight matrix of the residuals therefore is

$$W_{c_i c_i} \quad = \quad (B_i^{\mathsf{T}} \Sigma_{y_i y_i} B_i)^{-1} = W_{A_i^{\circ} A_i^{\circ}} \,. \tag{19.133}$$

The normal equation matrix thus is

$$N = \sum_i A_i^{\mathsf{T}} W_{c_i c_i} A_i \tag{19.134}$$

$$= \sum_i J_r^{\mathsf{T}} \begin{bmatrix} -\widehat{\boldsymbol{N}} & 0_{3\times 2} \\ (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0) & S(\widehat{\boldsymbol{N}}) M^{(s)\mathsf{T}} \end{bmatrix} W_{A_i^{\circ} A_i^{\circ}} \begin{bmatrix} -\widehat{\boldsymbol{N}}^{\mathsf{T}} & (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0)^{\mathsf{T}} \\ 0_{2\times 3} & -M^{(s)} S(\widehat{\boldsymbol{N}}) \end{bmatrix} J_r \tag{19.135}$$

$$= \sum_i \begin{bmatrix} \boldsymbol{q}_3^{\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{0}^{\mathsf{T}} & \boldsymbol{q}_1^{\mathsf{T}} \\ \boldsymbol{0}^{\mathsf{T}} & \boldsymbol{q}_2^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} -\widehat{\boldsymbol{N}} & 0_{3\times 2} \\ (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0) & S(\widehat{\boldsymbol{N}}) M^{(s)\mathsf{T}} \end{bmatrix} W_{A_i^{\circ} A_i^{\circ}} \begin{bmatrix} -\widehat{\boldsymbol{N}}^{\mathsf{T}} & (\widehat{\boldsymbol{X}}_{0i} - \widehat{\boldsymbol{X}}_0)^{\mathsf{T}} \\ 0_{2\times 3} & -M^{(s)} S(\widehat{\boldsymbol{N}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{q}_1 & \boldsymbol{q}_2 \end{bmatrix}$$

$$= \sum_i \begin{bmatrix} -1 & 0 & 0 \\ X_{0i}'' - X_0'' & 1 & 0 \\ Y_{0i}'' - Y_0'' & 0 & 1 \end{bmatrix} \begin{bmatrix} w_{q_i} & & \\ & w_{\phi_i} & \\ & & w_{\psi_i} \end{bmatrix} \begin{bmatrix} -1 & X_{0i}'' - X_0'' & Y_{0i}'' - Y_0'' \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{19.136}$$

$$= \sum_i \begin{bmatrix} -1 & 0 & 0 \\ X_{0i}'' - X_0'' & 1 & 0 \\ Y_{0i}'' - Y_0'' & 0 & 1 \end{bmatrix} \begin{bmatrix} -w_{q_i} & w_{q_i}(X_{0i}'' - X_0'') & w_{q_i}(Y_{0i}'' - Y_0'') \\ & w_{\phi_i} & \\ & & w_{\psi_i} \end{bmatrix} \tag{19.137}$$

$$= \sum_i \begin{bmatrix} w_{q_i} & -w_{q_i}(X_{0i}'' - X_0') & -w_{q_i}(Y_{0i}'' - Y_0'') \\ -w_{q_i}(X_{0i}'' - X_0'') & w_{\phi_i} + w_{q_i}(X_{0i}'' - X_0'')^2 & w_{q_i}(X_{0i}'' - X_0')(Y_{0i}'' - Y_0'') \\ -w_{q_i}(Y_{0i}'' - Y_0'') & w_{q_i}(X_{0i}'' - X_0'')(Y_{0i}'' - Y_0'') & w_{\psi_i} + w_{q_i}(Y_{0i}'' - Y_0'')^2 \end{bmatrix} \tag{19.138}$$

The normal equation matrix is diagonal, if[1]

$$\sum_i w_{q_i}(X_{0i}'' - X_0'') = \sum_i w_{q_i}(Y_{0i}'' - Y_0'') = 0 \,. \tag{19.139}$$

Then we obtain

$$N = \sum_i \begin{bmatrix} w_{q_i} & 0 & 0 \\ 0 & w_{\phi_i} + w_{q_i} X_{0i}'^2 & \\ 0 & & w_{\psi_i} + w_{q_i} Y_{0i}'^2 \end{bmatrix} . \tag{19.140}$$

Hence, all entities have to be taken at their estimates. If we use the centroid of the fitted centroids

$$\boldsymbol{X}_0'' = \frac{\sum_i w_{q_i} \widehat{\boldsymbol{X}}_{0i}''}{\sum_i w_{q_i}} \tag{19.141}$$

and the individual centroids reduced to the common centroid

$$\boldsymbol{X}_{0i}' = \boldsymbol{X}_{0i}'' - \boldsymbol{X}_0'' \,. \tag{19.142}$$

The right hand side of the normal equation system is

$$\boldsymbol{n} \quad = \quad \sum_i A_i^{\mathsf{T}} W_{c_i c_i} (-\boldsymbol{g}_i(\hat{\boldsymbol{\theta}}^a, \hat{\boldsymbol{y}}^a) + B_i(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i) \,. \tag{19.143}$$

[1] The original note said $\sum_i w_{\phi_i}(X_{0i}'' - X_0'') = \sum_i w_{\psi_i}(Y_{0i}'' - Y_0'') = 0$. But due to (19.141), this appears to be incorrect.

19.4.3.2 The Theoretical Precision

We obtain the variances for the three entities

$$\sigma_{\hat{q}}^2 = \frac{1}{I}\,\frac{1}{\overline{w_q}} \tag{19.144}$$

$$\sigma_{\hat{\phi}}^2 = \frac{1}{I}\,\frac{1}{\overline{w_\phi} + \overline{w_q}\,\overline{X_{0i}^2}} \tag{19.145}$$

$$\sigma_{\hat{\psi}}^2 = \frac{1}{I}\,\frac{1}{\overline{w_\psi} + \overline{w_q}\,\overline{Y_{0i}^2}}\,. \tag{19.146}$$

If the I planes would have the same precision we would obtain

$$\sigma_{\hat{q}}^2 = \frac{1}{I}\,\sigma_q^2\,, \quad \sigma_{\hat{\phi}}^2 = \frac{1}{I}\,\frac{\sigma_q^2\sigma_\phi^2}{\sigma_q^2 + \sigma_\phi^2\overline{X_{0i}^2}}\,, \quad \sigma_{\hat{\psi}}^2 = \frac{1}{I}\,\frac{\sigma_q^2\sigma_\psi^2}{\sigma_q^2 + \sigma_\psi^2\overline{Y_{0i}^2}}\,. \tag{19.147}$$

This is a plausible result: The precision of the normal of the average plane increases with the number I of the planes and with increasing scatter of the individual planes. Observe, if the standard deviation σ_q is 0, then the directions will also have standard deviation 0.

19.5 Motion from Plane to Plane correspondences

19.5.1 Problem Statement

Given are I correspondences $\{\mathcal{A}_i, \mathcal{A}'_j\}$ which are related by

$$\mathcal{M} : \mathcal{A}'_j \mapsto \mathcal{A}_i \qquad \mathcal{A}_i \equiv \mathcal{M}(\mathcal{A}'_j) \quad \text{for all } (ij) \in \mathcal{C}\,. \tag{19.148}$$

There are two options to establish the correspondences:

1. The planes $(\mathcal{A}_i, \mathcal{A}'_j)$ refer to the planar patches derived from some segmentation of two point clouds. Then each of the planes $\mathcal{A}_i$ or $\mathcal{A}'_j$ may have several correspondences, namely if there are coplanar planes one or both of the point clouds. The Jacobian B of the Gauss–Helmert model is block diagonal, each block B_k referring to the correspondence of coplanar planes $\{i_k\}$ and $\{j'_k\}$ in the two point clouds.
2. The planes $(\mathcal{A}_i, \mathcal{A}'_j)$ refer to aggregated coplanar planes in each point cloud. Then there is a one-to-one correspondence, and we may refer to the same index, thus refer to $(\mathcal{A}_k, \mathcal{A}'_k)$. In this case the partitioning of the point cloud has a final merge-step to find sets of coplanar points and to determine the average (ML-estimates) plane parameters.

We do not distinguish the two cases until we discuss the solution of the nonlinear Gauss–Helmert model.

We explicitly have

$$\mathcal{A}_i : \quad \left\{\boldsymbol{X}_0, \mathsf{Q}; \sigma_q^2, \sigma_\phi^2, \sigma_\psi^2\right\}_i \tag{19.149}$$

The constraint implies an unknown motion $\mathcal{M}$

$$\mathcal{M} : \quad \{\boldsymbol{T}, \mathsf{R}\} \tag{19.150}$$

which transforms the 3D points $\mathcal{X}_i$ into the coordinate system

$$\boldsymbol{X} = \mathsf{R}\boldsymbol{X}' + \boldsymbol{T}\,. \tag{19.151}$$

The corresponding transformation of the plane parameters is

$$\boldsymbol{X}_{0i} = \mathsf{R}\boldsymbol{X}'_{0j} + \boldsymbol{T} \tag{19.152}$$

and

$$Q_i = RQ'_j\,. \tag{19.153}$$

We need three constraints for the identity of two planes. These can be the following rotational constraint

$$\boxed{\boldsymbol{N}_i = R\boldsymbol{N}'_j\,,} \tag{19.154}$$

which represents two degrees of freedom, and the translational constraint

$$\boxed{\boldsymbol{N}_i^\mathsf{T}(R\boldsymbol{X}'_{0j} + \boldsymbol{T} - \boldsymbol{X}_{0i}) = 0\,,} \tag{19.155}$$

which represents the third degree of freedom.

From a counting argument we would need only two planes. However, then the translation along the intersecting 3D line is not determined. Therefore, we need at least three planes in general position for being able to determine the motion.

19.5.2 Minimal Solution for the Motion from Three Plane Correspondences

The three planes need to intersect in a 3D point $\mathcal{Y}$ not at infinity. Otherwise the translation in this direction is not determined.

Then the translation can be determined from the two intersection points $\boldsymbol{Y}$ and $\boldsymbol{Y}'$, and the rotation from the three normals.

If enough plane-plane correspondences are available the rotation may be derived from (19.154) in the form

$$N = RN' \tag{19.156}$$

Hence we have

$$H = N'^\mathsf{T}N = U\boldsymbol{\Lambda}V^\mathsf{T} \tag{19.157}$$

and thus

$$R = UV^\mathsf{T}\,. \tag{19.158}$$

Using this rotation the translation then can be determined from (19.155) in the form

$$\boldsymbol{N}_i^\mathsf{T}(R\boldsymbol{X}'_{0j} - \boldsymbol{X}_{0i}) = -\boldsymbol{N}_j^\mathsf{T}\boldsymbol{T} \tag{19.159}$$

which leads to the linear equation system

$$B^\mathsf{T}B\boldsymbol{T} = B^\mathsf{T}\boldsymbol{b} \tag{19.160}$$

with

$$B = -N = -[\boldsymbol{N}_i^\mathsf{T}] \quad \text{and} \quad \boldsymbol{b} = [\boldsymbol{N}_i^\mathsf{T}(R\boldsymbol{X}'_{0j} - \boldsymbol{X}_{0i})]\,. \tag{19.161}$$

Weighting is possible.

19.5.3 An Iterative Solution

We use the three constraints for each correspondence

$$\underset{3\times1}{\boldsymbol{g}_{ij}}(\hat{\boldsymbol{T}}, \hat{R}(\hat{\boldsymbol{\theta}}); \widehat{\boldsymbol{X}}_{0i}, \widehat{\boldsymbol{N}}_i, \widehat{\boldsymbol{X}}'_{0j}, \widehat{\boldsymbol{N}}'_j) = \begin{bmatrix} \widehat{\boldsymbol{N}}_i^\mathsf{T}(\hat{R}\widehat{\boldsymbol{X}}'_{0j} + \hat{\boldsymbol{T}} - \widehat{\boldsymbol{X}}_{0i}) \\ M_i^{(s)}S(\widehat{\boldsymbol{N}}_i)\hat{R}\widehat{\boldsymbol{N}}'_j \end{bmatrix} = \boldsymbol{0}\,. \tag{19.162}$$

Where $M_i^{(s)}S(\widehat{\boldsymbol{N}}_i) \in \mathsf{null}^\mathsf{T}(\boldsymbol{N}_i^\mathsf{T})$ is a orthonormal 2×3 matrix which is achieved by selecting two independent rows of the skew symmetric matrix $S(\widehat{\boldsymbol{N}}_i)$.

The linearized model reads as

$$\boldsymbol{g}_{ij}(\hat{\boldsymbol{T}}, \hat{R}(\hat{\boldsymbol{\theta}}); \widehat{\boldsymbol{X}}_{0i}, \widehat{\boldsymbol{N}}_i, \widehat{\boldsymbol{X}}'_{0j}, \widehat{\boldsymbol{N}}'_j) = \boldsymbol{g}_{ij}(\hat{\boldsymbol{\theta}}^a, \hat{\boldsymbol{y}}^a) \tag{19.163}$$
$$+ \begin{bmatrix} (\hat{R}\widehat{\boldsymbol{X}}'_{0j} + \hat{\boldsymbol{T}} - \widehat{\boldsymbol{X}}_{0i})^\mathsf{T}\widehat{\Delta\boldsymbol{N}}_i - \widehat{\boldsymbol{N}}_i^\mathsf{T}S(\hat{R}\widehat{\boldsymbol{X}}'_{0j})\widehat{\Delta\boldsymbol{\theta}} + \widehat{\boldsymbol{N}}_i^\mathsf{T}\hat{R}\widehat{\Delta\boldsymbol{X}}'_{0j} + \widehat{\boldsymbol{N}}_i^\mathsf{T}\widehat{\Delta\boldsymbol{T}} - \widehat{\boldsymbol{N}}_i^\mathsf{T}\widehat{\Delta\boldsymbol{X}}_{0i} \\ -M_i^{(s)}S(\hat{R}\widehat{\boldsymbol{N}}'_j)\widehat{\Delta\boldsymbol{N}}_i - M_i^{(s)}S(\widehat{\boldsymbol{N}}_i)S(\hat{R}\widehat{\boldsymbol{N}}'_j)\widehat{\Delta\boldsymbol{\theta}} + M_i^{(s)}S(\widehat{\boldsymbol{N}}_i)\hat{R}\widehat{\Delta\boldsymbol{N}}'_j \end{bmatrix}^a$$

Hence we have

$$\boldsymbol{g}_{ij} = \boldsymbol{g}_{ij}(\widehat{\boldsymbol{\theta}}^a, \widehat{\boldsymbol{y}}^a) + \mathsf{X}_{ij}\widehat{\Delta\boldsymbol{\theta}} + \mathsf{Z}_{ij}^{\mathsf{T}}\widehat{\Delta\boldsymbol{y}} = \mathbf{0} \tag{19.164}$$

with

$$\underset{6\times 1}{\Delta\boldsymbol{\theta}} := \begin{bmatrix} \Delta\boldsymbol{T} \\ \Delta\boldsymbol{\theta} \end{bmatrix} \quad \text{and} \quad \underset{6\times 1}{\Delta\boldsymbol{y}} := \begin{bmatrix} \Delta\boldsymbol{A}_i^\circ \\ \Delta\boldsymbol{A}_j^\circ \end{bmatrix} \tag{19.165}$$

The Jacobians are

$$\underset{3\times 6}{\mathsf{X}_{ij}^{\mathsf{T}}} = \frac{\partial \boldsymbol{g}_{ij}}{\partial\widehat{\boldsymbol{\theta}}} = \begin{bmatrix} -\widehat{\boldsymbol{N}}_i^{\mathsf{T}} \mathsf{S}(\widehat{R}\widehat{\boldsymbol{X}}_{0j}') & \widehat{\boldsymbol{N}}_i^{\mathsf{T}} \\ -M_i^{(s)}\mathsf{S}(\widehat{\boldsymbol{N}}_i)\mathsf{S}(\widehat{R}\widehat{\boldsymbol{N}}_j') & \mathsf{0} \end{bmatrix}^a \tag{19.166}$$

and

$$\underset{3\times 6}{\mathsf{Z}_{ij}^{\mathsf{T}}} = \frac{\partial \boldsymbol{g}_{ij}}{\partial[\boldsymbol{A}_i^{\mathsf{T}}, \boldsymbol{A}_j'^{\mathsf{T}}]^{\mathsf{T}}} \tag{19.167}$$

$$= \frac{\partial \boldsymbol{g}_{ij}}{\partial[\boldsymbol{A}_i^{*\mathsf{T}}, \boldsymbol{A}_j'^{*\mathsf{T}}]^{\mathsf{T}}} \frac{\partial[\boldsymbol{A}_i^{*\mathsf{T}}, \boldsymbol{A}_j'^{*\mathsf{T}}]^{\mathsf{T}}}{\partial[\boldsymbol{A}_i^{\mathsf{T}}, \boldsymbol{A}_j'^{\mathsf{T}}]^{\mathsf{T}}} \tag{19.168}$$

$$= \underset{3\times 12}{\begin{bmatrix} -\widehat{\boldsymbol{N}}_i^{\mathsf{T}} & (\widehat{R}\widehat{\boldsymbol{X}}_{0j}' + \widehat{\boldsymbol{T}} - \widehat{\boldsymbol{X}}_{0i})^{\mathsf{T}} & \widehat{\boldsymbol{N}}_i^{\mathsf{T}}\widehat{R} & \mathbf{0}^{\mathsf{T}} \\ \mathsf{0}_{2\times 3} & -M_i^{(s)}\mathsf{S}(\widehat{R}\widehat{\boldsymbol{N}}_j') & \mathsf{0}_{2\times 3} & M_i^{(s)}\mathsf{S}(\widehat{\boldsymbol{N}}_i)\widehat{R} \end{bmatrix}^a} \begin{bmatrix} J_r(Q_i) & \\ & J_r(Q_j') \end{bmatrix}^a \tag{19.169}$$

If each plane only is present in one constraint, hence we have $i = j$, the normal equations for the six unknown parameters read as

$$N\widehat{\Delta\boldsymbol{\theta}} = \boldsymbol{n} \tag{19.169}$$

with

$$\underset{6\times 6}{N} = \sum_i \underset{6\times 3}{A_i} (\underset{3\times 6}{B_i^{\mathsf{T}}} \underbrace{\mathrm{Diag}(\{\Sigma_{A_i^\circ A_i^\circ}, \Sigma_{A_i'^\circ A_i'^\circ}\})}_{6\times 6} \underset{6\times 3}{B_i})^{-1} \underset{3\times 6}{A_i^{\mathsf{T}}} \tag{19.170}$$

$$\underset{6\times 1}{\boldsymbol{n}} = \sum_i \underset{6\times 3}{A_i} (\underset{3\times 6}{B_i^{\mathsf{T}}} \underbrace{\mathrm{Diag}(\{\Sigma_{A_i^\circ A_i^\circ}, \Sigma_{A_i'^\circ A_i'^\circ}\})}_{6\times 6} \underset{6\times 3}{B_i})^{-1} \underbrace{(-\boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}^a, \widehat{\boldsymbol{y}}_i^a)) + B_i(\widehat{\boldsymbol{y}}_i^a - \boldsymbol{y}_i^a)}_{3\times 1} \tag{19.171}$$

The update of the translation and the rotation then is

$$\begin{bmatrix} \widehat{\boldsymbol{T}}^{(\nu+1)} \\ \widehat{R}^{(\nu+1)} \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{T}}^{(\nu)} + \widehat{\Delta\boldsymbol{T}} \\ R(\widehat{\Delta\boldsymbol{\theta}})\widehat{R}^{(\nu)} \end{bmatrix}. \tag{19.172}$$

19.5.4 Theoretical Accuracy of the Motion

We assume the rotation and translation is an identity. We also assume the corresponding planes to have the same mean parameters and the same covariance matrix. This simplifies the expressions and allows us to derive the covariance matrix as a function of the planes.

We use the relations

$$D_r = \boldsymbol{r}\boldsymbol{r}^{\mathsf{T}} \tag{19.173}$$

$$\mathsf{S}^2(\boldsymbol{r}) = -(I_3 - D_r) \tag{19.174}$$

$$\mathsf{S}(\boldsymbol{r})R = R\mathsf{S}(R^{\mathsf{T}}\boldsymbol{r}) \tag{19.175}$$

$$\boldsymbol{N} = Q\boldsymbol{e}_3 \quad \text{or} \quad \boldsymbol{e}_3 = Q^{\mathsf{T}}\boldsymbol{N}. \tag{19.176}$$

The Jacobians are (omitting the hats and assuming we always refer to the fitted values)

$$\mathsf{X}_i^{\mathsf{T}} = \frac{\partial \boldsymbol{g}_i}{\partial\widehat{\boldsymbol{\theta}}} = \begin{bmatrix} -\widehat{\boldsymbol{N}}_i^{\mathsf{T}}\mathsf{S}(\boldsymbol{X}_{0j}') & \boldsymbol{N}_i^{\mathsf{T}} \\ -M_i^{(s)}\mathsf{S}(\boldsymbol{N}_i)\mathsf{S}(\boldsymbol{N}_i') & \mathsf{0} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & \boldsymbol{N}_i^{\mathsf{T}} \\ M_i^{(s)}(I_3 - D_{N_i}) & \mathsf{0} \end{bmatrix} \tag{19.177}$$

and

$$Z_i^{\mathsf{T}} = \frac{\partial \boldsymbol{g}_{ij}}{\partial [\boldsymbol{A}_i^{\mathsf{T}}, \boldsymbol{A}_j'^{\mathsf{T}}]^{\mathsf{T}}} \tag{19.178}$$

$$= \begin{bmatrix} -\boldsymbol{N}_i^{\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} & \boldsymbol{N}_i^{\mathsf{T}} & \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{0}_{2\times 3} & -M_i^{(s)} S(\boldsymbol{N}_i) & \boldsymbol{0}_{2\times 3} & M_i^{(s)} S(\boldsymbol{N}_i) \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \boldsymbol{q}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{q}_{i1} & \boldsymbol{q}_{i2} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{bmatrix} \boldsymbol{q}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{q}_{i1} & \boldsymbol{q}_{i2} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} -\boldsymbol{e}_1^{\mathsf{T}} & \boldsymbol{e}_1^{\mathsf{T}} \\ -M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] & M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] \end{bmatrix} \tag{19.179}$$

We now assume the covariance matrices of all planes to be identical and isotropic

$$\Sigma_{A_i^\circ A_i^\circ} = \Sigma_{A^\circ A^\circ} = \begin{bmatrix} \sigma_q^2 & & \\ & \sigma_\phi^2 & \\ & & \sigma_\phi^2 \end{bmatrix} \tag{19.180}$$

Remark: Better do not do this! ◇

Then we have

$$B_i^{\mathsf{T}} \Sigma_{A_i^\circ A_i^\circ} B_i = \begin{bmatrix} -\boldsymbol{e}_1^{\mathsf{T}} & \boldsymbol{e}_1^{\mathsf{T}} \\ -M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] & M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] \end{bmatrix} \begin{bmatrix} \Sigma_{A^\circ A^\circ} & \\ & \Sigma_{A^\circ A^\circ} \end{bmatrix} \begin{bmatrix} -\boldsymbol{e}_1^{\mathsf{T}} & \\ -M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] & M_i^{(s)}[\boldsymbol{0}, \end{bmatrix}$$

$$= \begin{bmatrix} -\boldsymbol{e}_1^{\mathsf{T}} & \boldsymbol{e}_1^{\mathsf{T}} \\ -M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] & M_i^{(s)}[\boldsymbol{0}, \boldsymbol{q}_{i2}, -\boldsymbol{q}_{i1}] \end{bmatrix} \begin{bmatrix} -\Sigma_{A^\circ A^\circ} \boldsymbol{e}_3 & -\Sigma_{A^\circ A^\circ} \begin{bmatrix} \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{q}_{i2}^{\mathsf{T}} \\ -\boldsymbol{q}_{i1}^{\mathsf{T}} \end{bmatrix} M_i^{(s)\mathsf{T}} \\ \Sigma_{A^\circ A^\circ} \boldsymbol{e}_3 & \Sigma_{A^\circ A^\circ} \begin{bmatrix} \boldsymbol{0}^{\mathsf{T}} \\ \boldsymbol{q}_{i2}^{\mathsf{T}} \\ -\boldsymbol{q}_{i1}^{\mathsf{T}} \end{bmatrix} M_i^{(s)\mathsf{T}} \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_q^2 & \boldsymbol{0} \\ \boldsymbol{0} & 2\sigma_\phi^2 \, M_i^{(s)} (\boldsymbol{q}_{i1}\boldsymbol{q}_{i1}^{\mathsf{T}} + \boldsymbol{q}_{i2}\boldsymbol{q}_{i2}^{\mathsf{T}}) M_i^{(s)^{\mathsf{T}}} \end{bmatrix}$$

$$= 2\Sigma_{A^\circ A^\circ}$$

Hence the normal equation matrix is

$$N = \frac{1}{2} \sum_i \begin{bmatrix} \boldsymbol{X}_{0i} \times \boldsymbol{N}_i & (I_3 - D_{N_i}) M_i^{(s)\mathsf{T}} \\ \boldsymbol{N}_i & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} w_q & \\ & w_\phi I_2 \end{bmatrix} \begin{bmatrix} (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & \boldsymbol{N}_i^{\mathsf{T}} \\ M_i^{(s)} (I_3 - D_{N_i}) & \boldsymbol{0} \end{bmatrix}$$

$$= \frac{1}{2} \sum_i \begin{bmatrix} \boldsymbol{X}_{0i} \times \boldsymbol{N}_i & (I_3 - D_{N_i}) M_i^{(s)\mathsf{T}} \\ \boldsymbol{N}_i & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} w_q (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & w_q \boldsymbol{N}_i^{\mathsf{T}} \\ w_\phi M_i^{(s)} (I_3 - D_{N_i}) & \boldsymbol{0} \end{bmatrix} \tag{19.184}$$

$$= \frac{1}{2} \sum_i \begin{bmatrix} w_q D(\boldsymbol{X}_{0i} \times \boldsymbol{N}_i) + w_\phi (I_3 - D_{N_i}) M_i^{(s)\mathsf{T}} M_i^{(s)} (I_3 - D_{N_i}) & (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i) \boldsymbol{N}_i^{\mathsf{T}} \\ \boldsymbol{N}_i (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & w_q D(\boldsymbol{N}_i) \end{bmatrix}$$

$$= \frac{1}{2} \sum_i \begin{bmatrix} w_q D(\boldsymbol{X}_{0i} \times \boldsymbol{N}_i) + w_\phi (\boldsymbol{q}_{i1}\boldsymbol{q}_{i1}^{\mathsf{T}} + \boldsymbol{q}_{i2}\boldsymbol{q}_{i2}^{\mathsf{T}}) & w_q \, S(\boldsymbol{X}_{0i}) D(\boldsymbol{N}_i) \\ w_q \, D(\boldsymbol{N}) S(\boldsymbol{X}_{0i}) & w_q D(\boldsymbol{N}_i) \end{bmatrix} \tag{19.185}$$

$$= \frac{1}{2} \sum_i w_q \begin{bmatrix} (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)(\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i) \boldsymbol{N}_i \\ \boldsymbol{N}_i (\boldsymbol{X}_{0i} \times \boldsymbol{N}_i)^{\mathsf{T}} & \boldsymbol{N}_i \boldsymbol{N}_i^{\mathsf{T}} \end{bmatrix} + w_\phi \begin{bmatrix} \boldsymbol{q}_{i1}\boldsymbol{q}_{i1}^{\mathsf{T}} + \boldsymbol{q}_{i2}\boldsymbol{q}_{i2}^{\mathsf{T}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

or generally

$$\boxed{N = \frac{1}{2} \sum_i w_{q_i} \begin{bmatrix} \boldsymbol{X}_{0i} \times \boldsymbol{N}_i \\ \boldsymbol{N}_i \end{bmatrix} [\boldsymbol{X}_{0i} \times \boldsymbol{N}_i \,,\, \boldsymbol{N}_i] + w_{\phi_i} \begin{bmatrix} \boldsymbol{q}_{i1} \\ \boldsymbol{0} \end{bmatrix} [\boldsymbol{q}_{i1}^{\mathsf{T}} \,,\, \boldsymbol{0}^{\mathsf{T}}] + w_{\psi_i} \begin{bmatrix} \boldsymbol{q}_{i2} \\ \boldsymbol{0} \end{bmatrix} [\boldsymbol{q}_{i2}^{\mathsf{T}} \,,\, \boldsymbol{0}^{\mathsf{T}}]} \tag{19.186}$$

Reducing the parameters to the translation yields the reduced normal equation matrix

$$\overline{N}_{TT} = \frac{1}{2}\sum_i \left(w_{q_i} D(\boldsymbol{X}_{0i} \times \boldsymbol{N}_i) + w_{\phi_i}(\boldsymbol{q}_{i1}\boldsymbol{q}_{i1}^\mathsf{T} + \boldsymbol{q}_{i2}\boldsymbol{q}_{i2}^\mathsf{T})\right) \tag{19.187}$$

$$- \left(\sum_i w_{q_i}\ D(\boldsymbol{N}) S(\boldsymbol{X}_{0i})\right) \left(\sum_i w_{q_i} D(\boldsymbol{N}_i)\right)^{-1} \left(\sum_i w_{q_i}\ S(\boldsymbol{X}_{0i}) D(\boldsymbol{N}_i)\right) \tag{19.188}$$

which can be determined if

$$\sum_i w_{q_i} D(\boldsymbol{N}_i) = \sum_i w_{q_i} \boldsymbol{N}_i \boldsymbol{N}_i^\mathsf{T} \tag{19.189}$$

is regular: Therefore at least three planes with non-coplanar normals are necessary for a solution.

20 Planes from Points

We describe the statistically optimal estimation of a single and of multiple planes from a point cloud, where the full covariance matrix of all scene coordinates is available, e.g., from bundle adjustment. This procedure might be used to derive ground truth data for plane extraction or for homography estimation.

20.1 Preface

The note (2023) describes the statistically optimal estimation of a single and of multiple planes from a point cloud, where the full covariance matrix of the scene coordinates is available, e.g., from bundle adjustment. The solution for single planes differs from that of Sect. 19.3.2 in Ch. 19: There the plane is Euclideanly normalized, here they are spherically normalized, which leads to simpler expressions.

20.2 The Problem

Given are K sets $\{\{\mathcal{X}_i\}, i = i, ..., I\}_k, k = 1, ..., K$ of 3D points together with their complete covariance matrix $\Sigma = [\Sigma_{ik,ik}]$ the task is to finde the best fitting planes $\mathcal{A}_k$. We start with the derivation for a single plane and then generalize to multiple planes.

The motivation is to derive reference data for homographies for identified planes being seen in pairs of images, whose poses and scene points have been determined by bundle adjustment. Instead of including the plane constraints into the bundle adjustment, we propose to use the coordinates of the estimated scene points together with their full covariance matrix and determine the best fitting plane parameters. This can be seen as an estimation in steps (Kalman filtering) where in the second step the plane constraints are used to improve the estimates of the scene points, which in the first step have been determined without these constraints.

Though it is possible to estimate the planes individually, the resulting parameters are not optimal, since the mutual correlations between the scene points belonging to different planes are not taken into account.

We therefore just assume, the coordinates of the relevant scene points together with their full covariance matrix is available, e.g., when using the Ceres solver.

20.3 Formalization

We start with the case $K = 1$ and omit all indices referring tho the plane of interest.

20.3.1 The incidence constraint

We assume the points are given with their homogeneous coordinates $\mathbf{X}_i, i = 1, ..., I$ and their joint covariance matrix

$$\Sigma = [\Sigma_{\mathrm{X}_i\mathrm{X}_j}] = \left[\left[\begin{array}{cc} \Sigma_{X_iX_j} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{array} \right]\right], \quad \text{with} \quad i, j = 1, ..., I\,. \tag{20.1}$$

and the plane $\mathcal{A}$ is represented by its spherically normalized homogeneous coordinates $\mathbf{A}$ with

$$|\mathbf{A}| = 1\,. \tag{20.2}$$

The a point $\mathcal{X}_i$ lies on the plane $\mathcal{A}$ if

$$\mathbf{X}_i^\mathsf{T}\mathbf{A} = 0\,. \tag{20.3}$$

20.3.2 The optimization problem

We now want to optimally estimate the plane parameters. The observations and unknown parameters in a Gauss-Helmert model with constraints are

$$\underset{N\times 1}{\boldsymbol{y}} := [\mathbf{X}_i],, \quad \underset{4\times 1}{\boldsymbol{\theta}} := \mathbf{A} \quad \text{and} \quad \underset{N\times 1=4I\times 1}{\boldsymbol{y}} := \mathbb{E}(\underline{\boldsymbol{y}}) \tag{20.4}$$

For achieving a ML-estimation we want minimize the residuals $\boldsymbol{y} - \boldsymbol{l}$ squared and weighted with the full weight matrix W

$$\Omega(\boldsymbol{\theta}, \boldsymbol{y}) = (\boldsymbol{y} - \boldsymbol{y})^\mathsf{T} W (\boldsymbol{y} - \boldsymbol{y}) \quad \text{with} \quad W = \left[\left[\begin{array}{cc} \Sigma^{-1}_{X_iX_j} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{array} \right]\right] \tag{20.5}$$

subject to the constraints

$$\begin{array}{ll} \mathbf{0} = \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{y}) := & [\mathbf{y}_i^\mathsf{T}\boldsymbol{\theta}]\,, \\ \mathbf{0} = \boldsymbol{h}(\boldsymbol{\theta}) := & \frac{1}{2}(|\boldsymbol{\theta}|^2 - 1)\,. \end{array} \tag{20.6}$$

20.3.3 Conditioning and approximate values

We assume the following:

- We have conditioned the given coordinates

  $$\mathbf{X}_i^c = \mathsf{M}\mathbf{X}_i \quad \text{with} \quad \mathsf{M} = \left[\begin{array}{cc} \frac{1}{s} I_3 & -\frac{1}{s}\boldsymbol{\mu}_X \\ \mathbf{0}^\mathsf{T} & 1 \end{array} \right], \tag{20.7}$$

  and

  $$s = \sqrt{\frac{1}{3}\mathrm{tr}(\mathbb{C}\mathrm{ov}(\boldsymbol{X}_i)}\,, \quad \text{and} \quad \boldsymbol{\mu}_X = \frac{1}{I}\sum_i \boldsymbol{X}_i \tag{20.8}$$

  since in non-homogeneous coordinates we have $\boldsymbol{X}_i^c = (\boldsymbol{X}_i - \boldsymbol{\mu}_X)/s$. Hence, we have the conditioned covariance matrix

  $$\Sigma^c = [\mathsf{M}\Sigma_{ij}\mathsf{M}^\mathsf{T}] \tag{20.9}$$

  Since we determine the plane parameters $\hat{\boldsymbol{\theta}}^c = \hat{\mathbf{A}}^c$ in the conditioned coordinate system where we can uncondition the estimated plane parameters

  $$\hat{\boldsymbol{\theta}} = \mathsf{M}\hat{\boldsymbol{\theta}}^c \quad \text{since} \quad \mathbf{A}^c = \mathsf{M}^{-1}\mathbf{A}\,. \tag{20.10}$$

  together with their covariance matrix

  $$\Sigma_{\hat{\theta}\hat{\theta}} = \mathsf{M}\Sigma_{\hat{\theta}^c\hat{\theta}^c}\mathsf{M}^\mathsf{T} \tag{20.11}$$

- We have an approximate solution $\boldsymbol{\theta}^{c,a} := \mathbf{A}^{c,a}$ based on the conditioned 3D points assuming all have the same covariance matrix I_3.

20.3.4 The algorithm for estimating the parameters

We refer to PCV Sect. 8.3.2 and the note on the Gauss-Helmert model, Sect. 4.1 augmented by the constraints between the parameters. We omit all superscripts indicating that we have conditioned the data.

We start from the correlated observed I scene points in homogeneous coordinates $\{\boldsymbol{y}, \Sigma_{yy}\} := \{[\mathbf{X}_i], [\Sigma_{ij}]\}$, the constraints $\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{y}) := [\mathbf{y}_i^\mathsf{T}\mathbf{A}] = \mathbf{0}$ and $h(\boldsymbol{\theta}) = 1/2(|\boldsymbol{\theta}|^2 - 1)$, and the approximate values $\boldsymbol{\theta}^a := \mathbf{A}^a$ for the unknowns and $\boldsymbol{y}^a := [\mathbf{X}_i]$ for the mean observations. We obtain the following algorithm for an iterative solution:

1. Iterate until convergence

   (a) Determine the Jacobians X and Z at the current approximate values $(\boldsymbol{\theta}^a, \boldsymbol{y}^a)$. Here we have *Jacobians at current approximations*

   $$\underset{I\times 4}{X} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}} := Y^a = [\boldsymbol{y}_i^{a\mathsf{T}}], \quad \underset{I\times 4I}{Z^\mathsf{T}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}} = I_I \otimes \hat{\boldsymbol{\theta}}^{a\mathsf{T}} \quad \text{and} \quad \underset{1\times 4}{\boldsymbol{h}^\mathsf{T}} = \frac{\partial h}{\partial \boldsymbol{\theta}} := \boldsymbol{\theta}^{a,\mathsf{T}}. \tag{20.12}$$

   In the first iteration we have

   $$[\boldsymbol{y}_i^{(0)}] := [\mathbf{X}_i]. \tag{20.13}$$

   (b) Determine the contradictions $\boldsymbol{c}_g$ and $\boldsymbol{c}_h$ of the negative constraints at the approximate values $\boldsymbol{\theta}^a$ and $\boldsymbol{y}$ of the unknown parameters together with their weight matrix [1] *contradictions of constraints given the parameters*

   $$\underset{I\times 1}{\boldsymbol{c}_g} := -\underset{I\times 4}{[\boldsymbol{l}_i^\mathsf{T}]}\ \underset{4\times 1}{\boldsymbol{\theta}^a}, \quad \underset{I\times I}{W_{gg}} = (Z^\mathsf{T}\Sigma Z)^{-1} = \left(\left[\boldsymbol{\theta}^{a,\mathsf{T}}\Sigma_{ij}\boldsymbol{\theta}^a\right]\right)^{-1} \tag{20.14}$$

   and

   $$\underset{1\times 1}{c_h} = \frac{1}{2}(|\boldsymbol{\theta}^a|^2 - 1). \tag{20.15}$$

   (c) Solve the normal equation system for the corrections $\Delta\boldsymbol{\theta}$ and $\Delta\boldsymbol{y}$ of the parameters *normal equation system*

   $$\underbrace{\begin{bmatrix} X^\mathsf{T}W_{gg}X & \boldsymbol{h} \\ \boldsymbol{h}^\mathsf{T} & 0 \end{bmatrix}}_{N} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \mu \end{bmatrix} = \underbrace{\begin{bmatrix} X^\mathsf{T}W_{gg}\,\boldsymbol{c}_g \\ c_h \end{bmatrix}}_{\boldsymbol{m}}. \tag{20.16}$$

   (d) Update the approximate parameters

   $$\boldsymbol{\theta}^a := \mathsf{N}(\boldsymbol{\theta}^a + \Delta\boldsymbol{\theta}) \quad \text{with} \quad \mathsf{N}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{|\boldsymbol{x}|}. \tag{20.17}$$

   (e) Determine the corrections for the mean observations

   $$\Delta\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}^a - \Sigma(I_I \otimes \boldsymbol{\theta}^{a,\mathsf{T}})W_{gg}\boldsymbol{g}(\hat{\boldsymbol{\theta}}^a\boldsymbol{y}). \tag{20.18}$$

   (f) Update the approximate mean observations *update of approximate mean observations*

   $$\boldsymbol{y}^a := [\mathsf{N}^e(\boldsymbol{y}_i + \Delta\boldsymbol{y}_i)] \quad \text{with} \quad \mathsf{N}^e(\mathbf{X}) = \frac{\mathbf{X}}{X_4}. \tag{20.19}$$

2. Set the final estimates of the unknown parameters and of the mean observations, sometimes called the fitted observation $\hat{\boldsymbol{y}} := \hat{\boldsymbol{y}}$ *final estimates*

   $$\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}^a \quad \text{and} \quad \hat{\boldsymbol{y}} = \boldsymbol{y}^a. \tag{20.20}$$

3. Determine the estimated variance factor

estimated variance factor

$$\hat{\sigma}_0^2 = \frac{\boldsymbol{c}_g^\mathsf{T} W_{c_g c_g} \boldsymbol{c}_g}{I-4} \,. \tag{20.21}$$

If the model holds its expectation is equal to 1.

Observe: Instead of minimizing the squared residuals $\boldsymbol{y}-\boldsymbol{l}$ weighted with W in (20.5), thus minimize $||\boldsymbol{y}-\boldsymbol{y}||_W$, we equivalently may minimize the weighted residuals of the squared constraints $\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{y})$ weighted with their weight matrix $W_{c_g c_g}$, thus minimizing $||\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{y})||_{W_{c_g c_g}}$, in both cases taking the constraints (20.6) into account.

4. Determine the covariance matrix of the estimated parameters

covariance matrix of the estimated parameters

$$\begin{bmatrix} X^\mathsf{T} W_{gg} X & \boldsymbol{h} \\ \boldsymbol{h}^\mathsf{T} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_{\hat{\theta}\hat{\theta}} & \cdot \\ \cdot & \cdot \end{bmatrix} . \tag{20.22}$$

Remark: If the observational noise is small and an approximate solution is acceptable, the steps 1.(e–f) can be omitted. Then the Jacobians X and Z are to be determined at $(\boldsymbol{\theta}^a, \boldsymbol{y})$ instead of at $(\boldsymbol{\theta}^a, \boldsymbol{y}^a)$. ◇

The complete procedure is given in the algorithm below.

Algorithm 3: Plane from correlated points, assuming conditioned values.
$[\hat{\mathbf{A}}, \Sigma_{\hat{A}\hat{A}}, \hat{\sigma}_0^2, R] =$ `CorrelatedPoints2Plane_D`$([\mathbf{X}_i], [\Sigma_{ij}], \mathbf{A}^a, T_\theta,$ `maxiter`$)$
Input: observed values $\boldsymbol{y} = [\boldsymbol{y}_i] := [\mathbf{X}_i]$, full covariance matrix $\Sigma = [\Sigma_{ij}]$
approximate values $\mathbf{A}^a$,
parameters T_θ, `maxiter` for controlling convergence.
Output: estimated parameters $\hat{\mathbf{A}}, \Sigma_{\hat{A}\hat{A}}$ for plane, variance factor $\hat{\sigma}_0^2$, redundancy R.

Redundancy $R = I - 3$;
if $R < 0$ **then stop**, not enough constraints;
Iteration $\nu = 0$, approx. values $\hat{\boldsymbol{\theta}}^a := \mathbf{A}^a$, $\boldsymbol{y}^a := [\mathbf{X}_i]$, stopping variable: $s = 0$;
repeat
  Jacobians: : $A = [\boldsymbol{y}_i^{a,\mathsf{T}}]$, $\boldsymbol{h} = \hat{\boldsymbol{\theta}}^a$;
  Constraints: $\boldsymbol{c}_g = -[\boldsymbol{y}_i^\mathsf{T}]\boldsymbol{\theta}^a$, $c_h = -1/2(|\boldsymbol{\theta}^a|^2 - 1)$;
  Weight matrix of constraints: $W_{gg} = [\boldsymbol{\theta}^{a,\mathsf{T}} \Sigma_{ij} \boldsymbol{\theta}^a]^{-1}$;
  Build normal equation system: $[N, \boldsymbol{m}]$, see (20.16);
  if N is singular **then stop**: normal equation matrix is singular;
  Updates of parameter vector $\boldsymbol{\theta}^a := \mathsf{N}(\boldsymbol{\theta}^a + \Delta\boldsymbol{\theta})$;
  Corrections for fitted observations: $\Delta\boldsymbol{y}$, see (20.18);
  Update fitted observations $\boldsymbol{y}^a = [\mathsf{N}^e(\boldsymbol{y}_i^a + \Delta\boldsymbol{y}_i)]$, see (20.19);
  Set iteration: $\nu := \nu + 1$;
  if $\max_u(|\widehat{\Delta\theta}_u|/\sigma_{\hat{\theta}_u}^a) < T_\theta$ or $\nu =$ `maxiter` **then** $s = 2$;
until $s \equiv 2$;
Estimated parameters $\hat{\mathbf{A}} := \hat{\boldsymbol{\theta}}^a$ and covariance matrix : $\Sigma_{\hat{A}\hat{A}}$, see (20.22);
if $R > 0$ **then** variance factor $\hat{\sigma}_0^2 = \boldsymbol{c}_g^\mathsf{T} W_{gg} \boldsymbol{c}_g / R$;
else $\hat{\sigma}_0^2 = 1$;

20.4 Multiple planes

We generalize the solution to the case of simultaneously estimating a set of K planes, in order to exploit all information for one bundle adjustment. This will yield different results due to the correlation between the scene points.

[1] We do not indicate, that $\boldsymbol{c}_g$ depends on approximate values thus omit a superscript ^{a}.

We consider K planes $\Pi_k, k = 1, ..., K$ with their I_k points $\mathcal{X}_{ik}, (ik) \in \mathcal{I}_k$. We assume the point sets for different planes are disjunct. We collect the I_k homogeneous coordinates of the observed scene points and their expectation for plane k in the $I_k \times 4$ matrices

$$\underset{I_K \times 4}{\mathsf{X}_k} = [\mathbf{X}_{ik}^\mathsf{T}] \quad \text{and} \quad \mathsf{Y}_k = \mathbb{E}(\mathsf{X}_k)\,. \tag{20.23}$$

Then we have the following

$$G = \sum_k I_k \tag{20.24}$$

constraints

$$\underset{G \times 1}{\boldsymbol{g}} = [\boldsymbol{g}_{ik}] = [\mathbb{E}(\mathsf{X}_k)\mathbf{A}_k] = [\mathbb{E}(\mathbf{X}_{ik}^\mathsf{T})\mathbf{A}_k] = \mathbf{0}\,, \quad h_k = \frac{1}{2}(|\mathbf{A}_k|^2 - 1) = 0 \quad k = 1, ..., K\,. \tag{20.25}$$

Witht the $4K$ unknown parameters, the $4G$ observations and their expectations

$$\underbrace{\boldsymbol{x}}_{4K \times 1} = [\boldsymbol{x}_k] := [\mathbf{A}_k]\,, \quad \underbrace{\boldsymbol{y}}_{\sum_k I_k} = [\boldsymbol{y}_{ik}] = [\mathbf{X}_{ik}] \quad \text{and} \qquad \text{and} \quad \boldsymbol{y} = \text{vec}(\mathsf{Y}^\mathsf{T}) = [\mathbb{E}(\mathbf{X}_{ik})] \tag{20.26}$$

the Jacobians X and Z are the following using the approximate values for $\boldsymbol{\theta}$ and Y

$$\mathsf{X} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \text{Diag}([\mathsf{X}_k]) := \text{Diag}([\mathsf{Y}_k]) \quad \text{and} \quad \mathsf{Z}^\mathsf{T} = \text{Diag}([\mathsf{Z}_{ik}^\mathsf{T}]) := \text{Diag}([\hat{\boldsymbol{\theta}}_k^{a\mathsf{T}}]) \tag{20.27}$$

The Jacobian for the constraints is

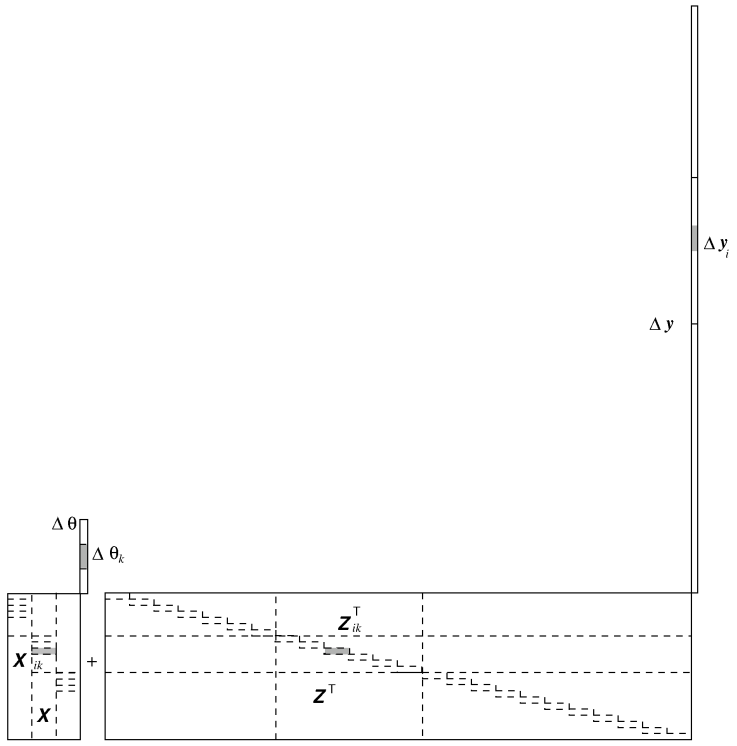


Figure 20.1: linearized constraints

$$\underset{4K \times K}{\mathsf{H}} = \text{Diag}([\boldsymbol{\theta}_k^a]) \tag{20.28}$$

Hence, with the approximate residuals

$$\boldsymbol{v}^a = \boldsymbol{y}^a - \boldsymbol{y} \tag{20.29}$$

we have the linearized optimization problem: Minimize

$$\Omega(\Delta\boldsymbol{\theta}, \Delta\boldsymbol{y}) = (\boldsymbol{y}^a + \Delta\boldsymbol{y})^\mathsf{T} W^+ (\boldsymbol{y}^a + \Delta\boldsymbol{y}) \quad \text{with} \quad W = \left[\left[\begin{array}{cc} \Sigma^{-1}_{X_i X_j} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{array}\right]\right] \tag{20.30}$$

subject to the constraints

$$\begin{array}{ll} \mathbf{0} = \boldsymbol{g}(\Delta\boldsymbol{\theta}, \Delta\boldsymbol{y}) := & A\Delta\boldsymbol{\theta} + Z^\mathsf{T}\Delta\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{\theta}^a, \boldsymbol{y}^a)\,, \\ \mathbf{0} = \boldsymbol{h}(\Delta\boldsymbol{\theta}) := & H^\mathsf{T}\Delta\boldsymbol{\theta} - \boldsymbol{h}(\Delta\boldsymbol{\theta}^a)\,. \end{array} \tag{20.31}$$

The full weight matrix of the constraints is

$$\underset{G\times G}{W_{gg}} = \left(\left[\boldsymbol{\theta}^{a,\mathsf{T}}_{ik} \Sigma_{ik,i'k'} \boldsymbol{\theta}^a_{i'k'}\right]\right)^{-1} \quad \text{with} \quad (ik) \in \mathcal{I}_k, k = 1, ..., K \tag{20.32}$$

hence, with the residual constraints

$$\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{\theta}^a, \boldsymbol{y}) = -[A_k]\boldsymbol{\theta}^a \quad \text{and} \quad \boldsymbol{c}_h = -\boldsymbol{h}(\boldsymbol{\theta}^a) \tag{20.33}$$

the normal equation system is

$$\underbrace{\left[\begin{array}{cc} A^\mathsf{T} W_{gg} A & H \\ H^\mathsf{T} & 0 \end{array}\right]}_{\underset{5K\times 5K}{N}} \left[\begin{array}{c} \underset{4K\times 1}{\Delta\boldsymbol{\theta}} \\ \underset{K\times 1}{\boldsymbol{\mu}} \end{array}\right] = \underbrace{\left[\begin{array}{c} A^\mathsf{T} W_{gg}\, \boldsymbol{c}_g \\ \boldsymbol{c}_h \end{array}\right]}_{\boldsymbol{m}}. \tag{20.34}$$

which, except for the block off-diagonal matrix H, is full. The algorithm above requires transparent adaptions.

Observe, the resulting plane parameters will be mutually correlated. But their individual 4×4 covariance matrix $\mathbb{D}(\underline{\mathbf{A}_k})$ may be reported as uncertainty of the *ground truth*.

20.5 Outlier detection

It may be useful to eliminate individual scene points before a final plane estimation. The following test statistic can be used for outlier detection

$$X_{ik} = \boldsymbol{c}^\mathsf{T}_{g_{ik}} W_{g_{ik},g_{ik}} \boldsymbol{c}_{g_{ik}} = w_{g_{ik},g_{ik}} \mathbf{X}^\mathsf{T}_{ik} \hat{\boldsymbol{\theta}}_k\,. \tag{20.35}$$

hence we explicitly need the weight matrix W_{gg} in (20.32). If the given model is correct, especially if the covariance matrix of the scene points is correct, then the test statistic X_{ik} follows a χ^2_4-distribution.

In case, we normalize the test statistic by some estimate for the variance factor, its distribution is not known.

21 Direct Solutions for the Similarity from Plane Pairs

We collect some direct solutions for determining the similarity (or motion) from corresponding plane pairs, representing point clouds. Some of the solutions are able to handle the case, where the sign of the normals are not consistent.

21.1 Problem

Given are plane pairs $\{\mathbf{A}_i, \mathbf{A}'_i\}_i = 1, ..., I$ which are assumed to be related by the similarity

$$\mathbf{A}'_i = \mathsf{H}^{-\mathsf{T}} \mathbf{A}_i \,. \tag{21.1}$$

determine a good estimate of H

$$\mathsf{H} = \begin{bmatrix} \lambda R & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} R & \boldsymbol{T}/\lambda \\ \mathbf{0}^\mathsf{T} & 1/\lambda \end{bmatrix} . \tag{21.2}$$

We assume the planes to be Euclideanly normalized

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{N} \\ -S \end{bmatrix} , \qquad \text{with} \qquad |\boldsymbol{N}| = 1. \tag{21.3}$$

In addition, we assume the coordinates to be conditioned, i. e. the distances of the planes to the origin should be less than 1. This can be achieved by a proper similarity transformation of coordinate system, such that the origin is in the center of all points and the distances S_i have absolute coordinates less than 1.

As the normals may not be consistent, as $\mathbf{A}$ and $-\mathbf{A}$ represent the same plane, we can distinguish two types of solutions, one which assumes the normals to be consistent, the other assuming they are not consistent.

In the following we first discuss solutions which do not exploit the full covariance structure or even do not refer to a statistical description of the uncertainty.

21.2 Minimal solutions

We discuss minimal a minimal solution for spatial similarity and for spatial motion.

Sect.	No. I of planes	normals	reflection
21.2.1	$I = 4$	consistent	allowed
21.2.2	$I = 3$	consistent	allowed
21.3.1	$I \geq 4$	not consistent	allowed
21.3.2	$I \geq 4$	consistent	allowed
		not consistent	not allowed

Table 21.1: Direct solution for the similarity from plane pairs

21.2.1 A one-step direct solution of a similarity from four plane pairs

The direct solution can be obtained from $\mathsf{H}^\mathsf{T}\mathbf{A}'_i = \mathbf{A}_i$ or $\mathbf{A}_i{}'^\mathsf{T}\mathsf{H} = \mathbf{A}_i^\mathsf{T}$, or

$$\mathsf{A} = \begin{bmatrix} \mathbf{A}_1^\mathsf{T} \\ \mathbf{A}_2^\mathsf{T} \\ \mathbf{A}_2^\mathsf{T} \\ \mathbf{A}_2^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{A}'^\mathsf{T}_1 \\ \mathbf{A}'^\mathsf{T}_2 \\ \mathbf{A}'^\mathsf{T}_3 \\ \mathbf{A}'^\mathsf{T}_4 \end{bmatrix} \begin{bmatrix} R & \boldsymbol{T}/\lambda \\ \mathbf{0}^\mathsf{T} & 1/\lambda \end{bmatrix} = \mathsf{A}'\mathsf{H} \tag{21.4}$$

Thus we directly obtain

$$\mathsf{H} = (\mathsf{A}')^{-1}\mathsf{A} \tag{21.5}$$

The matrix would be the correct result, if the data were noiseless. This is valid for both, a similarity and a motion.

Therefore, in general we enforce the matrix to be a similarity by enforcing the upper left 3×3-matrix to be a rotation and the lack of a projective component. With

$$\mathsf{H}(1:3,1:3) = UDV^\mathsf{T} \tag{21.6}$$

we therefore have the best estimate for a *similarity*

$$\mathsf{H} = \begin{bmatrix} |D|^{1/3}UV^\mathsf{T} & \mathsf{H}(1:3,4) \\ \mathbf{0}^\mathsf{T} & \mathsf{H}(4,4) \end{bmatrix} \tag{21.7}$$

This solution assumes the normals of the planes to be consistent. It allows for a mirroring.

21.2.2 A two-step solution for a motion from three planes

The two-step solution first determines the rotation from the three normals and then the translation from the intersection point.

Rotation. The rotation directly can be determined from the normals using

$$B' = [\boldsymbol{N}'_1, \boldsymbol{N}'_2, \boldsymbol{N}'_3] = R[\boldsymbol{N}_1, \boldsymbol{N}_2, \boldsymbol{N}_3] = RB \tag{21.8}$$

from

$$R = B^{-1}B' \tag{21.9}$$

which in case the data are noisy is no rotation. The best rotation is again obtained from the SVD of $R = UDV^\mathsf{T}$ from

$$\hat{R} = UV^\mathsf{T} \tag{21.10}$$

If the data are related by a reflection, then $\det(\hat{R}) = -1$.

Translation. The translation can easily be determined from the intersection point of the three planes.

Also, this solution assumes the normals of the planes to be consistent. The result allows the data to contain a reflection.

21.3 Direct solutions the similarity from $I \geq 4$ plane pairs

21.3.1 One step procedure

The basic constraint for each plane can be written as (see Heuel 2004, eq. (3.29) and sect. 3.3.1.6, tables 3.5 and 3.9)

$$\mathcal{A}_i \equiv \mathcal{A}_i' : \ \mathbf{A}_i \cap (\mathsf{H}^\mathsf{T}\mathbf{A}_i') = \overline{\Pi}(\mathbf{A}_i)\mathsf{H}^\mathsf{T}\mathbf{A}_i' = \mathbf{0}\,, \tag{21.11}$$

or

$$(\overline{\Pi}(\mathbf{A}_i) \otimes \mathbf{A}_i^\mathsf{T})\mathrm{vec}\mathsf{H} \stackrel{!}{=} \mathbf{0} \tag{21.12}$$

with the matrix

$$\underset{6\times4}{\overline{\Pi}(\mathbf{A})} = \begin{bmatrix} S(\boldsymbol{N}) & \mathbf{0} \\ -SI_3 & -\boldsymbol{N} \end{bmatrix} \tag{21.13}$$

containing the skew matrix $S(\boldsymbol{N})$ of the 3-vector $\boldsymbol{N}$. Observe, this constraint is independent on the sign of the plane vectors.

This gives rise to the direct solution

$$\underbrace{\begin{bmatrix} \overline{\Pi}(\mathbf{A}_1) \otimes \mathbf{A}_1^\mathsf{T} \\ \cdots \\ \overline{\Pi}(\mathbf{A}_i) \otimes \mathbf{A}_i^\mathsf{T} \\ \cdots \\ \overline{\Pi}(\mathbf{A}_I) \otimes \mathbf{A}_I^\mathsf{T} \end{bmatrix}}_{\underset{6I\times16}{B}} \mathbf{h} \stackrel{!}{=} \mathbf{0} \tag{21.14}$$

The best estimate for $\mathbf{h}$ is the right singular vector of the $6I \times 16$-matrix B belonging to the smallest singular value.

As each plane gives rise to three constraints, we need at least five planes. As we know that the elements $\mathsf{H}_{4,1:3}$ are zero, we can cancel the corresponding columns in the matrix B, then being of size $6I \times 12$ and can do with four planes minimum.

The result is an affinity

$$\mathsf{H} = \begin{bmatrix} A & \boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & s \end{bmatrix} \tag{21.15}$$

which needs to be enforced to become a similarity, with

$$A = UDV^\mathsf{T} \tag{21.16}$$

leading to

$$\hat{\mathsf{H}} = \begin{bmatrix} |D|^{1/3}/s\, UV^\mathsf{T} & \boldsymbol{T}/s \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{21.17}$$

Since only the deviation from the 0-constraints (21.14) is minimized, this solution allows the normals to be inconsistent. Again, if the data contain a reflection, the solution will be a reflection.

21.3.2 Two step procedure

We first determine the rotation, then rotate the planes and then determine translation and scale. Thus we assume the similarity to be

$$\hat{\mathsf{H}} = \begin{bmatrix} \hat{R} & \hat{\boldsymbol{T}}/\hat{\lambda} \\ \mathbf{0} & 1/\hat{\lambda} \end{bmatrix} = \begin{bmatrix} I_3 & \hat{\boldsymbol{T}}' \\ \mathbf{0}^\mathsf{T} & \hat{\mu}' \end{bmatrix} \begin{bmatrix} \hat{R} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{21.18}$$

with

$$\hat{\boldsymbol{T}} = \hat{\boldsymbol{T}}'/\hat{\mu}'\,, \qquad \hat{\lambda} = 1/\hat{\mu}' \tag{21.19}$$

21.3.2.1 Determining the rotation

Assuming consistency of the normals. For finding the optimal rotation we minimize the optimization function

$$\sum_i p_i |\boldsymbol{N}'_i - R\boldsymbol{N}_i|^2 \tag{21.20}$$

which is equivalent to maximize

$$\sum_i p_i \boldsymbol{N}'_i R \boldsymbol{N}_i = \mathrm{tr}(RH)\,, \qquad \mathsf{H} = \sum_i p_i \boldsymbol{N}_i \boldsymbol{N}'_i \tag{21.21}$$

The weights p_i can be approximated by

$$p_i = \frac{1}{\sigma^2_{\phi_i} + \sigma^2_{\phi'_i}} \approx \frac{N_i^3 N_i'^3}{N_i^3 + N_i'^3}\,. \tag{21.22}$$

The approximation is valid in case the planes have been determined from N_i and N'_i points, assuming the normals to have isotropic uncertainty. The solution can be found by using the SVD (or equivalently using quaternions)

$$H = UDV^\mathsf{T} \tag{21.23}$$

leading to the rotation

$$R = VU^\mathsf{T}\,. \tag{21.24}$$

If the data contain a reflection, then $\det R = -1$.

Not assuming consistency of the normals. From the constraints

$$\mathcal{N}'_i \equiv \mathcal{R}(\mathcal{N}_i):\ \boldsymbol{N}'_i \times R\boldsymbol{N}_i = S(\boldsymbol{N}'_i)R\boldsymbol{N}_i = (\boldsymbol{N}_i^\mathsf{T} \otimes S(\boldsymbol{N}'_i))\mathrm{vec}R \stackrel{!}{=} \mathbf{0} \tag{21.25}$$

we obtain the joint constraints

$$\begin{bmatrix} \boldsymbol{N}_1^\mathsf{T} \otimes S(\boldsymbol{N}'_1) \\ \cdots \\ \boldsymbol{N}_i^\mathsf{T} \otimes S(\boldsymbol{N}'_i) \\ \cdots \\ \boldsymbol{N}_I^\mathsf{T} \otimes S(\boldsymbol{N}'_I) \end{bmatrix} \boldsymbol{r} \stackrel{!}{=} \mathbf{0} \tag{21.26}$$

This yields an approximation for a rotation matrix, except for the sign. Hence, we are not able to allow for reflections. From $R = UDV^\mathsf{T}$ we obtain an estimate for the rotation

$$\widehat{R} = UV^\mathsf{T}\mathrm{sign}(|UV'|) \tag{21.27}$$

with $\det \widehat{R} = 1$.

21.3.2.2 Rotating the planes

We now rotate the planes, which just needs to be applied to the normals, therefore

$$\overline{\mathbf{A}}_i = \begin{bmatrix} \overline{\boldsymbol{N}}_i \\ -S_i \end{bmatrix} = \begin{bmatrix} \widehat{R}\boldsymbol{N}_i \\ -S_i \end{bmatrix}, \qquad \overline{\mathbf{A}}'_i = \begin{bmatrix} \overline{\boldsymbol{N}}'_i \\ -S'_i \end{bmatrix} = \begin{bmatrix} \widehat{R}\boldsymbol{N}'_i \\ -S'_i \end{bmatrix} \tag{21.28}$$

These planes only differ by scale and translation.

21.3.2.3 Estimating translation and scale

Transforming planes by translation $\boldsymbol{T}'$ and scale μ' is performed by

$$\overline{\mathbf{A}}_i = \begin{bmatrix} I_3 & \mathbf{0} \\ \boldsymbol{T}'^{\mathsf{T}} & \mu' \end{bmatrix} \overline{\mathbf{A}}'_i \tag{21.29}$$

thus only refers to the distances S_i and S'_i. We have the constraint

$$c_i = S_i - [\overline{\boldsymbol{N}}_i'^{\mathsf{T}} - S'_i] \begin{bmatrix} \boldsymbol{T}' \\ \mu \end{bmatrix} \stackrel{!}{=} 0 \tag{21.30}$$

with an approximate weight

$$w_i \approx \frac{1}{\sigma_{q_i}^2 + \sigma_{q'_i}^2} \approx \frac{N_i N'_i}{N_i + N'_i} \tag{21.31}$$

for the uncertainty of the position across the planar patches (but see the critics below).

Therefore, we can determine the scale and the translation from

$$\boldsymbol{S} = \begin{bmatrix} S_1 \\ \dots \\ S_i \\ \dots \\ S_I \end{bmatrix} \stackrel{!}{=} \underbrace{\begin{bmatrix} \overline{\mathbf{A}}'^{\mathsf{T}}_1 \\ \dots \\ \overline{\mathbf{A}}'^{\mathsf{T}}_i \\ \dots \\ \overline{\mathbf{A}}'^{\mathsf{T}}_I \end{bmatrix}}_{\substack{B \\ I \times 4}} \begin{bmatrix} \boldsymbol{T}' \\ \mu' \end{bmatrix} \tag{21.32}$$

The least squares solution for the translation and the scale is

$$\begin{bmatrix} \widehat{\boldsymbol{T}}' \\ \widehat{\mu}' \end{bmatrix} = (B^{\mathsf{T}} W B)^{-1} B W^{\mathsf{T}} \boldsymbol{S}\,, \qquad W = \mathrm{Diag}([w_1, ..., w_i, ..., w_I])\,. \tag{21.33}$$

which in the case of four planes reduces to

$$\begin{bmatrix} \widehat{\boldsymbol{T}}' \\ \widehat{\mu}' \end{bmatrix} = B^{-1} \boldsymbol{S} \tag{21.34}$$

The procedure cannot be based on some statistical model.

21.4 Stability of the solution

In case all planes are parallel the rotation cannot be determined.

In case the normals $\boldsymbol{A}_{hi}$ of the planes are coplanar, the translation cannot be determined.

In case the four planes intersect in one point the four plane vectors are linearly dependent and the matrices A and A' in (21.4) are singular or - in case of noise - close to singular. Then the scale cannot be determined.

In case the normals are well distributed the condition numbers

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{21.35}$$

of A and A' should be significantly less than the inverse standard deviation of the directions measured in radiants.

22 A Permutation Invariant Test Statistic for the Circularity of Four Points

We propose a statistical test for evaluating whether four points are co-circular. The test is invariant to the numbering of the points. It can be used to decide, whether an edge in a Delaunay triangulation is certain or uncertain.

22.1 Preface

This note (1998) provides a permutation invariant test statistic for four point lying on a circle. It tests the imaginary part $\Im(c(z_1, z_2, z_3, z_4))$ of the cross ratio of the four points in the complex plane, which needs to be real, due to a circle-preserving Möbius transformation onto the real axis. The note was the basis for Förstner (1999)

22.2 The Problem

Checking the stability of the neighbourhoods of geometric features, especially points, derived from a Voronoi diagram or a Delaunay-Triangulation (cf. Fig. 22.1) can use the geometric configuration of the four points causing the endpoints of each edge PQ: the two points B and C of the two neighbouring Voronoi cells and the two points A and D neighboured to these two points.

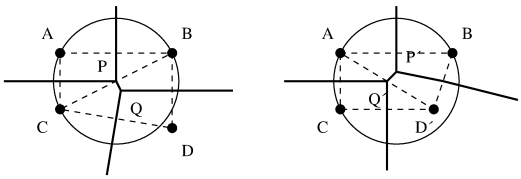


Figure 22.1: Two alternative Voronoi Diagrams for four nearly cocircular points

In case Point D is slightly shifted to D' the edge PQ will disappear and change into the edge $P'Q'$, indicating A and D' to be neighboured.

The transition appears when D passes the circle through (ABC). Therefore it is reasonable to check the closeness of the four points to a circle.

The goal of this note is to develop a test statistic for testing the hypothesis four points lying on a circle.

22.3 The Test Statistic

Given the planar coordinates (x_i, y_i) of the four points $P_i, i = 1, 2, 3, 4$ collected in complex numbers

$$z_i = x_i + jy_i \tag{22.1}$$

the four points ly on a circle in case the cross ratio

$$c(z_1, z_2, z_3, z_4) = \frac{z_1 - z_4}{z_2 - z_4} : \frac{z_1 - z_3}{z_2 - z_3} \tag{22.2}$$

is real, or if

$$t = \Im(c(z_1, z_2, z_3, z_4)) = 0 \tag{22.3}$$

The proof of this result uses the fact that a homography $z' = (a + bz)/(c + dz)$ of the complex plane, the Möbius-Transformation, is circle preserving. Therefore, a homography can be found which maps a circle to a straight line, say the real line, the cross ratio of any four points on it is real.

Assuming the given points are uncertain with covariance matrix $\boldsymbol{C}$ which in the simplest case could be $\boldsymbol{C} = \sigma^2 \boldsymbol{I}$ one can derive the standard deviation σ_t of t and obtain the test statistic

$$T = \frac{t}{\sigma_t} \sim N(0, 1) \tag{22.4}$$

in case one can assume Gaussian distribution of the given points.

22.4 The Invariance of the Test Statistic

The independence of the distribution of the test statistic results in the invariance on the numbering of the points.

22.5 Numerical example

Given the four equally distant points (0,0), (1,0), (2,0) and (3,0), the cross ratio is 3/4, indicating them to be collinear.

If the fourth point has coordinates $(3, s)$, the cross ratio is

$$c(z_1, z_2, z_3, z_4) = \frac{1}{2}\,\frac{3 + j\,s}{2 + j\,s} \tag{22.5}$$

its imaginary part is

$$t = \Im(c(z_1, z_2, z_3, z_4)) = -\frac{1}{2}\,\frac{s}{4 + s^2} \tag{22.6}$$

Assuming the points to be uncertain by σ in all coordinates, the variance of t is given by:

$$\sigma_t^2 = \frac{1}{8}\,\frac{3\,s^4 + 29\,s^2 + 40}{\left(4 + s^2\right)^2}\,\sigma^2 \tag{22.7}$$

as the Jacobian of t is

$$\boldsymbol{J}_{cx} = \left(\frac{\partial c}{\partial x_i}\right) = \frac{1}{4}\begin{pmatrix} \frac{s}{\sqrt{4+s^2}} \\ \frac{2+s^2}{\sqrt{4+s^2}} \\ -8\,\frac{s}{(4+s^2)^2} \\ -2\,\frac{12+9\,s^2+s^4}{(4+s^2)^2} \\ -\frac{s}{\sqrt{4+s^2}} \\ \frac{s^2+6}{\sqrt{4+s^2}} \\ 8\,\frac{s}{(4+s^2)^2} \\ 2\,\frac{-4+s^2}{(4+s^2)^2} \end{pmatrix} \tag{22.8}$$

The test statistic is

$$T(s) = \frac{t}{\sigma_t} = \frac{t}{\sigma\sqrt{\boldsymbol{J}_{cx}^T\boldsymbol{J}_{cx}}} = \frac{s}{\sigma}\,\frac{-\sqrt{2}}{\sqrt{3\,s^4+29\,s^2+40}}\,. \tag{22.9}$$

For small values t the expression nearly is linear in t, see Fig. 22.2 Thus, for small s we

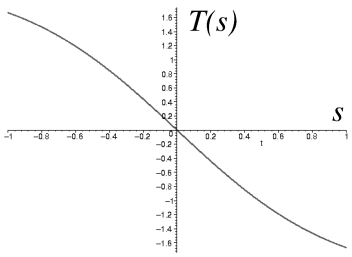


Figure 22.2: Test statistic $T(s)$

have:

$$T(s) = \frac{t(s)}{\sigma_t(s)} \approx \frac{-s/8}{\sigma\sqrt{5/16}} = -\frac{s}{\sigma\sqrt{20}} \tag{22.10}$$

where $-s/8$ is the linear approximation of the imaginary part of the cross ratio (cf. eq. 22.6) and

$$\sigma_t^2 = \frac{5}{16}\sigma^2 \tag{22.11}$$

has been found by error propagation, as

$$\boldsymbol{J}_{cx} = \left(\frac{\partial c}{\partial x_i}\right) = \frac{1}{8}\begin{pmatrix} 0 \\ 1 \\ 0 \\ -3 \\ 0 \\ 3 \\ 0 \\ -1 \end{pmatrix} \tag{22.12}$$

If we now exchange the second and the third point, thus $\{(0,0),(2,0),(1,0),(3,s)\}$, we obtain the cross ratio

$$c'(z_1,z_2,z_3,z_4') = -\frac{3+j\,s}{1+j\,s} \tag{22.13}$$

which is -3 for $s = 0$, compared to $3/4$ before. The imaginary part is

$$t' = 2\,\frac{s}{1+s^2} \tag{22.14}$$

With the Jacobian

$$\boldsymbol{J}'_{cx} = \left(\frac{\partial c}{\partial x_i}\right) = \frac{1}{4}\begin{pmatrix} \frac{s}{1+s^2} \\ -\frac{2+s^2}{1+s^2} \\ \frac{s\left(7+3\,s^2\right)}{\left(1+s^2\right)^2} \\ -\frac{3\,s^2+s^4+6}{\left(1+s^2\right)^2} \\ 4\,\frac{s}{1+s^2} \\ 2\,\frac{s^2+3}{1+s^2} \\ -4\,\frac{s}{\left(1+s^2\right)^2} \\ -2\,\frac{-1+s^2}{\left(1+s^2\right)^2} \end{pmatrix} \overset{s=0}{=} \begin{pmatrix} 0 \\ -2 \\ 0 \\ -6 \\ 0 \\ 6 \\ 0 \\ 2 \end{pmatrix} \tag{22.15}$$

we obtain the variance of t'

$$\sigma_t'^2 = 2\,\frac{3\,s^4+29\,s^2+40}{\left(1+s^2\right)^2} \tag{22.16}$$

which is different than before exchanging points 2 and 3. We no have – except for the sign – the same test statistic

$$T'(s) = \frac{t'}{\sigma'_t} = \frac{t'}{\sigma\sqrt{\boldsymbol{J}_{cx}^T\boldsymbol{J}_{cx}}} = \frac{s}{\sigma}\,\frac{\sqrt{2}}{\sqrt{3\,s^4+29\,s^2+40}} \tag{22.17}$$

specializing to

$$T(s) = \frac{t'(s)}{\sigma'_t(s)} \approx \frac{2s}{\sigma\sqrt{80}} = \frac{s}{\sigma\sqrt{20}} \tag{22.18}$$

for small s.

23 Uncertain Ellipses

We discuss various representations for uncertain ellipses, since they regularly occur as perspective images of 3D circles. We treat them as special conics, and provide some of their relations. Especially we propose two minimal representations for ellipses.

23.1 Preface

The note (2014) discusses various representations for ellipses, as special conics, and provides some of their relations. Especially it contains two minimal representations. We discuss how to represent their uncertainty using a covariance matrix of the parameters.

23.2 Task and problem

There are several possibilities to represent uncertain ellipses:

1. The centroid form fixes the centre, the semiaxes and the direction of the major semi-axis:
$$\boldsymbol{e} = [x_0, y_0, a, b, \phi]\,, \quad \text{with} \quad a > 0, b > 0\,, \qquad \Sigma_{ee}\,. \tag{23.1}$$
It indicates, that an ellipse has 5 degrees of freedom.
This representation is useful for plotting. In case $a = b$, when the ellipse is a circle, the angle ϕ is not required.

2. The homogeneous 3×3-matrix
$$\mathsf{C} = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} = \begin{bmatrix} \mathsf{C}_{hh} & \boldsymbol{c}_{h0} \\ \boldsymbol{c}_{0h} & c_{00} \end{bmatrix}. \tag{23.2}$$
together with its partitioning into homogeneous and non-homogeneous part. This representation does not enforce the curve $qx^\mathsf{T}\mathsf{C}\mathbf{x} = 0$ to be an ellipse. It is used for geometric reasoning. Its uncertainty is represented using its vector:

3. The 9-vector
$$\mathbf{c} = \mathrm{vec}\mathsf{C} \tag{23.3}$$
is used for representing the matrix as vector. It allows to give the 9×9-covariance matrix Σ_{cc} of the elements of C, which ideally should have rank 5.

4. The non-normalized 6-vector $\mathbf{c} = \mathrm{vech}\mathsf{C}$

$$\mathbf{c}^h = [a, b, c, d, e, f]^\mathsf{T}\,, \qquad \Sigma_{\mathbf{c}^h\mathbf{c}^h} \tag{23.4}$$

is a minimal *homogeneous* representation. Again, the covariance matrix should have the rank 5.

5. The *spherically* normalized 6-vector $\mathbf{c}^s$

$$\mathbf{c}^s = \frac{\mathbf{c}}{|\mathbf{c}|}\,, \qquad \Sigma_{\mathbf{c}^s\mathbf{c}^s} \tag{23.5}$$

using the constraint $|\mathbf{c}^s| = 1$ or

$$g^s(\mathbf{c}^s) = \frac{1}{2}(|\mathbf{c}^s|^2 - 1) = 0\,. \tag{23.6}$$

It often is the basic result of some algebraic estimation. Due to the constraint, the covariance matrix has rank 5.
Still, the representation does not enforce the conic to be an ellipse.

6. The *spectrally* normalized 6-vector $\mathbf{c}^\sigma$

$$\mathbf{c}^\sigma = \frac{\mathbf{c}}{\sqrt{|\mathsf{C}_{hh}|}} = [a^\sigma, b^\sigma, c^\sigma, d^\sigma, e^\sigma, f^\sigma]^\mathsf{T}\,, \qquad \Sigma_{\mathbf{c}^\sigma\mathbf{c}^\sigma}\,. \tag{23.7}$$

using the constraint

$$g^\sigma(\mathbf{c}^\sigma) = a^\sigma c^\sigma - (b^\sigma)^2 - 1 = 0\,. \tag{23.8}$$

It often is the basic result of an algebraic solution. Again, due to the constraint, the covariance matrix has rank 5.
Only conics with $|\mathsf{C}_{hh}| > 0$, thus ellipses, can be normalized this way. During estimation the constraint enforces the conic to be an ellipse.
It scales the conic matrix differently than the spherical normalization: for small positive $|\mathsf{C}_{hh}|$ the elements of C may become large, indicating the distinction from a parabola is difficult.
The constraint does not fix the sign of the conic. A similar constraint $a^\sigma c^\sigma - (b^\sigma)^2 + 1 = 0$ could be used to enforce hyperbolae.

7. The *reduced* 5-vector $\boldsymbol{c}^r$

$$\boldsymbol{c}^r = \mathsf{J}_r^\mathsf{T}\,\mathbf{c}^s \quad \text{with} \quad \mathsf{J}_r(\mathbf{c}) = \mathsf{null}(\mathbf{c}^\mathsf{T})\,, \qquad \Sigma_{\mathbf{c}^r\mathbf{c}^r} \tag{23.9}$$

lying in the tangent space of $\mathbf{c}^s$ allowing a ML-estimation without constraints. The Jacobian J_r is the null space of the Jacobian $\partial g^s/\partial \mathbf{c}^s$ of the constraint (23.6).
It is a minimal representation, why $\Sigma_{\mathbf{c}^r\mathbf{c}^r}$ in general has full rank.
This representation is valid for general conics, not only for ellipses. Therefore, given some approximate values an iterative estimation procedure may change the type of conic, unless care is taken when updating the parameters during the iteration, e.g., enforcing the update to keep the constraint using the exponential map of C_{hh}.
This is the reason to possibly use the following representation if ellipses are of concern.

8. The 5-vector $\boldsymbol{c}^m$ with minimal parametrization

$$\boldsymbol{c}^m = [b^\sigma, c^m, d^\sigma, e^\sigma, f^\sigma]^\mathsf{T} \quad \text{with} \quad c^m = \frac{1 - (a^\sigma)^2 + (b^\sigma)^2}{2a^\sigma}\,, \qquad \Sigma_{\mathbf{c}^m\mathbf{c}^m} \tag{23.10}$$

can be used to generate a spectrally normalized conic matrix such that

$$\mathsf{C}^\sigma = \begin{bmatrix} \sqrt{1 + (b^\sigma)^2 + (c^m)^2} - c^m & b^\sigma & d^\sigma \\ b^\sigma & \sqrt{1 + (b^\sigma)^2 + (c^m)^2} + c^m & e^\sigma \\ d^\sigma & e^\sigma & f^\sigma \end{bmatrix} \tag{23.11}$$

and determinant

$$|C^{\sigma}_{hh}| = \begin{vmatrix} \sqrt{1+(b^{\sigma})^2+(c^m)^2}-c^m & b^{\sigma} \\ b^{\sigma} & \sqrt{1+(b^{\sigma})^2+(c^m)^2}+c^m \end{vmatrix} = 1\,. \quad (23.12)$$

It can be used for ML-estimation without additional constraints. In contrast to the minimal representation with reduced coordinates $\boldsymbol{c}^r$ it can only be used for ellipses. The normalization of the parameters leads to large elements if the ellipse is close to a parabola.
A similar construction can be developed for hyperbola.

In the following we collects some relations.

23.3 Selected relations

23.3.1 Transformation between c and $\mathbf{c}^h$

The vector $\mathbf{c}^h$ is derived from the vector $\mathbf{c}$ by selection. We explicitly have

$$\mathbf{c}^h = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = S_{69} \begin{bmatrix} a \\ b \\ d \\ b \\ c \\ e \\ d \\ e \\ f \end{bmatrix} = S_{69}\mathbf{c} \quad (23.13)$$

with

$$S_{69} = \frac{\partial \mathbf{c}^h}{\partial \mathbf{c}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (23.14)$$

with

$$S_{69}S_{69}^{\mathsf{T}} = I_6 \quad \text{and} \quad S_{69}^{\mathsf{T}}S_{69} = \text{Diag}\left([1,1,1,0,1,1,0,0,1]\right)\,. \quad (23.15)$$

The inverse relation is the filling of c

$$\mathbf{c} = \begin{bmatrix} a \\ b \\ d \\ b \\ c \\ e \\ d \\ e \\ f \end{bmatrix} = S_{96} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = S_{96}\mathbf{c}^h \quad (23.16)$$

with

$$S_{96} = \frac{\partial \mathbf{c}}{\partial \mathbf{c}^h} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23.17}$$

with

$$S_{96}S_{96}^\mathsf{T} = I_9 \quad \text{and} \quad S_{96}^\mathsf{T}S_{96} = \text{Diag}\left([1,2,1,2,2,1]\right) . \tag{23.18}$$

We have the relation between the two Jacobians

$$S_{69}S_{96} = I_6 , \qquad S_{96}S_{69} = I_9 \tag{23.19}$$

23.3.2 Transformation from $\mathbf{c}^h$ to $\mathbf{c}^s$

The spherical normalization has the Jacobian

$$J_{sh} = \frac{\partial \mathbf{c}^s}{\partial \mathbf{c}^h} = \frac{1}{|\mathbf{c}^h|}(I_6 - \mathbf{c}^s\mathbf{c}^{s\mathsf{T}}) . \tag{23.20}$$

The first factor reflects the scaling, the second factor reflects the constraint. Therefore

$$J_{sh}\mathbf{c} = \mathbf{0} \quad \text{and} \quad J_{sh}J_{sh} = \frac{1}{|\mathbf{c}^h|}J_{sh} . \tag{23.21}$$

Thus, because of $\mathbf{c}^\mathsf{T}J_{sh}^\mathsf{T} = \mathbf{0}$, the Jacobian $J_{sh} = \mathsf{null}(\mathbf{c}^\mathsf{T})$ is the null space of $\mathbf{c}^\mathsf{T}$, and, except for a factor, is idempotent.

As $\mathbf{c}^\sigma$ is a special case of $\mathbf{c}^h$, the Jacobian J_{sh} can also be used for a transformation from $\mathbf{c}^\sigma$ to $\mathbf{c}^s$:

$$J_{s\sigma} = \frac{\partial \mathbf{c}^s}{\partial \mathbf{c}^\sigma} = \frac{1}{|\mathbf{c}^\sigma|}(I_6 - \mathbf{c}^s\mathbf{c}^{s\mathsf{T}}) . \tag{23.22}$$

The inverse relation, i.e., the derivation of $\mathbf{c}^h$ can be simplified to be the identity.

23.3.3 Transformation from $\mathbf{c}^h$ to $\mathbf{c}^\sigma$

Given $\mathbf{c}^h$ the Jacobian

$$J_{\sigma h} = \frac{\partial \mathbf{c}^\sigma}{\partial \mathbf{c}^h} \tag{23.23}$$

results from (23.7). With $D = a^h c^h - (b^h)^2$ have

$$\begin{aligned} \sqrt{D}\,\mathbf{c}^\sigma &= \mathbf{c}^h && (23.24)\\ \mathrm{d}\sqrt{D}\,\mathbf{c}^\sigma + \sqrt{D}\,\mathrm{d}\mathbf{c}^\sigma &= \mathrm{d}\mathbf{c}^h && (23.25)\\ \mathbf{c}^\sigma\,\frac{1}{2}\frac{1}{\sqrt{D}}\mathrm{d}D + \sqrt{D}\,\mathrm{d}\mathbf{c}^\sigma &= \mathrm{d}\mathbf{c}^h && (23.26) \end{aligned}$$

Now

$$\mathrm{d}D = [c^h, -2b^h, a^h, 0, 0, 0]\mathrm{d}\mathbf{c}^h . \tag{23.27}$$

Therefore

$$\mathbf{c}^\sigma\,\frac{1}{2}\frac{1}{\sqrt{D}}[c^h, -2b^h, a^h, 0, 0, 0]\mathrm{d}\mathbf{c}^h + \sqrt{D}\,\mathrm{d}\mathbf{c}^\sigma = \mathrm{d}\mathbf{c}^h \tag{23.28}$$

Solving for $\mathrm{d}\mathbf{c}^\sigma$

$$\mathrm{d}\mathbf{c}^\sigma = \frac{1}{\sqrt{D}}\mathrm{d}\mathbf{c}^h - \frac{1}{2\sqrt{D}}\mathbf{c}^\sigma[c^h, -2b^h, a^h, 0, 0, 0]\mathrm{d}\mathbf{c}^h \tag{23.29}$$

$$\mathrm{d}\mathbf{c}^\sigma = \frac{1}{\sqrt{D}}\left[I_6 - \frac{1}{2}\mathbf{c}^\sigma[c^h, -2b^h, a^h, 0, 0, 0]\right]\mathrm{d}\mathbf{c}^h \tag{23.30}$$

Thus

$$J_{\sigma h} = \frac{\partial \mathbf{c}^\sigma}{\partial \mathbf{c}^h} = \frac{1}{\sqrt{D}}\left[I_6 - \frac{1}{2\sqrt{D}}\mathbf{c}^h[c^h, -2b^h, a^h, 0, 0, 0]\right]. \tag{23.31}$$

In case the homogeneous vector already is spectrally normalized, such that $D = 1$, then

$$J_{\sigma h} = \frac{\partial \mathbf{c}^\sigma}{\partial \mathbf{c}^h} = \left[I_6 - \frac{1}{2}\mathbf{c}^h[c^h, -2b^h, a^h, 0, 0, 0]\right]. \tag{23.32}$$

23.3.4 Transformation between $\mathbf{c}^\sigma$ and $\mathbf{c}^m$

The two vectors have the last three elements in common. Therefore we only need to address the first elements. Let

$$\boldsymbol{d}^\sigma = \begin{bmatrix} a^\sigma \\ b^\sigma \\ c^\sigma \end{bmatrix} \quad \text{and} \quad \boldsymbol{d}^m = \begin{bmatrix} b^m \\ c^m \end{bmatrix}. \tag{23.33}$$

The we have the relation

$$\boldsymbol{d}^\sigma = \begin{bmatrix} \sqrt{1 + (b^m)^2 + (c^m)^2} - c^m \\ b^m \\ \sqrt{1 + (b^m)^2 + (c^m)^2} + c^m \end{bmatrix} \quad \text{and} \quad \boldsymbol{d}^m = \begin{bmatrix} b^\sigma \\ \dfrac{1 - (a^\sigma)^2 + (b^\sigma)^2}{2a^\sigma} \end{bmatrix} \tag{23.34}$$

This yields the Jacobians

$$\underset{3\times 2}{J_{d^\sigma d^m}} = \frac{\partial \boldsymbol{d}^\sigma}{\partial \boldsymbol{d}^m} = \begin{bmatrix} \dfrac{b^m}{\sqrt{1 + (b^m)^2 + (c^m)^2}} & \dfrac{c^m}{\sqrt{1 + (b^m)^2 + (c^m)^2}} - 1 \\ 1 & 0 \\ \dfrac{b^m}{\sqrt{1 + (b^m)^2 + (c^m)^2}} & \dfrac{c^m}{\sqrt{1 + (b^m)^2 + (c^m)^2}} + 1 \end{bmatrix} \tag{23.35}$$

and

$$\underset{2\times 3}{J_{d^m d^\sigma}} = \frac{\partial \boldsymbol{d}^m}{\partial \boldsymbol{d}^\sigma} = \begin{bmatrix} 0 & 1 & 0 \\ -\dfrac{1 + (a^\sigma)^2 + (b^\sigma)^2}{2(a^\sigma)^2} & \dfrac{b^\sigma}{a^\sigma} & 0 \end{bmatrix} \tag{23.36}$$

with

$$J_{d^m d^\sigma} J_{d^\sigma d^m} = I_2 \quad \text{and} \quad J_{d^\sigma d^m} J_{d^m d^\sigma} = \begin{bmatrix} I_2 & \mathbf{0} \\ \left[-\dfrac{c^\sigma}{a^\sigma}, -\dfrac{b^\sigma}{a^\sigma}\right] & 0 \end{bmatrix} \tag{23.37}$$

Thus we have the Jacobians

$$\underset{6\times 5}{J_{c^\sigma c^m}} = \frac{\partial \boldsymbol{c}^\sigma}{\partial \boldsymbol{c}^m} = \begin{bmatrix} J_{d^\sigma d^m} & 0 \\ 0 & I_3 \end{bmatrix} \quad \text{and} \quad \underset{5\times 6}{J_{c^m c^\sigma}} = \frac{\partial \boldsymbol{c}^m}{\partial \boldsymbol{c}^\sigma} = \begin{bmatrix} J_{d^m d^\sigma} & 0 \\ 0 & I_3 \end{bmatrix} \tag{23.38}$$

Observe, we have the Jacobian of the constraint (23.8)

$$J_{g^\sigma c^\sigma} = \frac{\partial g^\sigma}{\partial \mathbf{c}^\sigma} = [c^\sigma, -2b^\sigma, a^\sigma, 0, 0, 0]. \tag{23.39}$$

and

$$J_{g^\sigma c^\sigma} J_{c^\sigma c^m} = \mathbf{0}_{5\times 1}. \tag{23.40}$$

Therefore the covariance matrix

$$\Sigma_{c^\sigma c^\sigma} = J_{c^\sigma c^m} \Sigma_{c^m c^m} J_{c^\sigma c^m}^\mathsf{T} \tag{23.41}$$

is singular and has null space $J_{g^\sigma c^\sigma}^\mathsf{T}$, as to be expected.

23.4 Open Problem

Though the uncertainty might be represented by a covariance matrix, this representation is only valid for relatively small standard deviations, since for larger deviations the parameters may represent a hyperbola. Fig. 23.1. The confidence regions are determined by simulating a large sample of 2D points, accumulating the resulting conics and thresholding the empirical density. The visualization of the uncertainty of ellipses is an open problem.

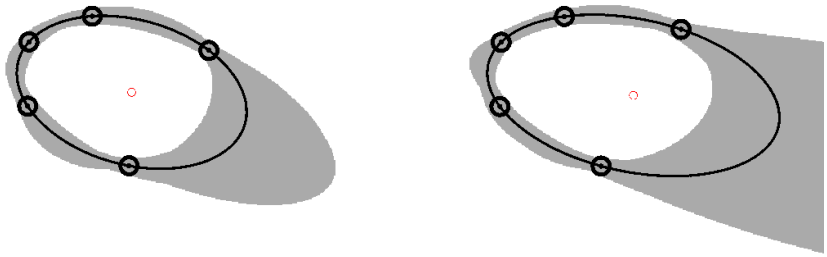


Figure 23.1: Ellipses with confidence regions. A slight change, here the vertical coordinate of the right most point, may lead to a dramatic change if the critical region

24 Minimal homography and its uncertainty from point pairs

The note provides an example for the uncertainty of minimal solutions, namely the determination of a homography from four corresponding points. It provides the covariance matrix of the resulting homography for both cases, where only one set of coordinates is uncertain and where both sets of coordinates are uncertain. An algorithm including the conditioning of the points is given.

24.1 Goal

We want to construct a 2D homography from a minimal set of point correspondences, i. e. a homography H from four corresponding 2D points $(\mathbf{p}_i, \mathbf{q}_i), i = 1, 2, 3, 4$ fulfilling:

$$\mathbf{q}_i = \mathsf{H}\mathbf{p}_i \,. \tag{24.1}$$

This is a special case for a minimal geometric problem.

We first provide a general method to derive the uncertainty of the derived parameters, following Barath et al. (2020b). For the example of a minimal solution for a homography from point pairs, we first discuss the case, where only one of the two sets of coordinates is uncertain and then generalize to the case where both sets of coordinates are uncertain.

24.2 Covariance matrix for the result of minimal problems

The task is to derive the U unkown parameters $\boldsymbol{\theta}$ from N observations $\boldsymbol{y}$ based on two sets of constraints: (1) a set of G constraints $\boldsymbol{g}$ between the unknown parameters $\boldsymbol{\theta}$ and (2) a set of $U - G$ constraints $\boldsymbol{h}$ only between the parameters, together collected in the form

$$\boldsymbol{k}(\boldsymbol{\theta}, \boldsymbol{y}) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{y}) \\ \boldsymbol{h}(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} . \tag{24.2}$$

The task is to solve for the parameters. The solution may be as simple as in our example, namely determining the nullspace of a matrix or arbitrarily complex, as for minimal solutions involving the essential matrix, generally leading to a set of T parameters $\boldsymbol{\theta}_t$ fulfilling the constraints.

We assume the observations are a sample from a multivariate distribution with second moments Σ_{yy}. Since the solutions are T functions

$$\boldsymbol{\theta}_t = \boldsymbol{f}_t(\boldsymbol{y}) \tag{24.3}$$

the uncertainty of the observations transfers to the uncertainty of the parameters $\boldsymbol{\theta}_t$.

In case, especially, if the relative accuracy of the observations is high, as in many geometric task, we may use variance propagation, neglecting higher order terms in a Taylor series of the function $\boldsymbol{f}_t$. In most cases this variance propagation is cumbersome. However, the derivation of the covariance matrices $\Sigma_{\theta_t\theta_t}$ may use implicit variance propagation, just based on the linearizations of the constraints, aas shown in the following.

The total differential of the constraints is given by

$$\underset{U\times U}{A_t}\mathrm{d}\boldsymbol{\theta}_t + \underset{U\times N}{B_t^\mathsf{T}}\mathrm{d}\boldsymbol{y} = \mathbf{0}\,, \quad \text{with} \quad \underset{U\times U}{A_t} = \left.\frac{\partial \boldsymbol{k}(\boldsymbol{\theta}_t, \boldsymbol{y})}{\partial \boldsymbol{\theta}_t}\right|_{\theta_t, y} \quad \text{and} \quad \underset{U\times N}{B_t^\mathsf{T}} = \left.\frac{\partial \boldsymbol{k}(\boldsymbol{\theta}_t, \boldsymbol{y})}{\partial \boldsymbol{y}}\right|_{\theta_t, y}\,. \tag{24.4}$$

If we assume small random perturbations in $\boldsymbol{\theta}$ and $\boldsymbol{y}$ we can solve for the perturbations $\Delta\boldsymbol{\theta}$ as a function of the perturbations $\Delta\boldsymbol{y}$:

$$\Delta\boldsymbol{\theta}_t = -A_t^{-1}B_t^\mathsf{T}\Delta\boldsymbol{y}\,. \tag{24.5}$$

Hence we directly obtain the covariance matrix of the parameters

$$\Sigma_{\theta_t\theta_t} = A_t^{-1}B_t^\mathsf{T}\Sigma_{yy}B_tA_t^{-\mathsf{T}}\,. \tag{24.6}$$

We demonstrate this result for the task of homography estimation from four point correspondences. We discuss the case, where only the transformed points are uncertain, and then generalize to the case where both point sets are uncertain and possibly correlated.

24.3 Only x is uncertain

We collect the unknown parameters in the vector

$$\mathbf{h} = \mathrm{vec}(\mathsf{H})\,. \tag{24.7}$$

We assume the homogeneous coordinates $\mathbf{p}_i$ to be given and only the homogeneous coordinates $\mathbf{q}_i$ of the transformed four points, collected in the 12-vector

$$\boldsymbol{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \mathbf{q}_4 \end{bmatrix}\,, \tag{24.8}$$

to be uncertain. Their covariance matrix is given by a 12×12 matrix Σ_{qq}.

In case, we only have given the non-homogeneous coordinates, then we use e.g.,

$$\mathbf{q}_i = \begin{bmatrix} \boldsymbol{q}_i \\ 1 \end{bmatrix} \quad \text{and} \quad \Sigma_{\mathbf{q}_i\mathbf{q}_i} = \begin{bmatrix} \Sigma_{q_iq_i} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}\,. \tag{24.9}$$

Then, we can rewrite the mapping (24.1) as three constraints

$$\mathbf{0} = \mathbf{q}_i \times \mathsf{H}\mathbf{p}_i = \mathsf{S}(\mathbf{q}_i)\mathsf{H}\;\mathbf{p}_i = -\mathsf{S}(\mathsf{H}\mathbf{p}_i)\;\mathbf{q}_i = \left(\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}(\mathbf{q}_i)\right)\;\mathbf{h}\,, \tag{24.10}$$

where the skew matrix $\mathsf{S}(\mathbf{q}) = \mathbf{q}_{[\times]}$ of the 3-vector $\mathbf{q}$ induces the cross product. The three constraints containing a skew matrix implicitely provide the Jadobians of the constraints w.r.t. the given points, the observed points annd the un known parameters.

Due to the rank two of the skew symmetric matrix, only two of the three constriants are independent. For the practical solution, one exploits the fact, that the constraints are trilinear, i.e., linear in all elements. If the we exclude points at infinity, we can reduce the three constraints by selecting only the first two, and arrive at the constrains betwen the parameters and the observations

$$\boldsymbol{g}_i(\mathbf{h}, \boldsymbol{q}) = \mathsf{S}^{(l)}(\mathbf{q}_i)\;\;\mathsf{H}\;\;\mathbf{p}_i = -\mathsf{S}^{(l)}(\mathsf{H}\mathbf{p}_i)\;\mathbf{q}_i = (\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}^{(l)}(\mathbf{q}_i)\;\mathbf{h} = \mathbf{0} \quad \text{with} \quad i = 1, 2, 3, 4\,, \tag{24.11}$$

with:

$$\mathsf{S}^{(l)}(\mathbf{x}) = [I_1 \mid \mathbf{0}]\, \mathsf{S}(\mathbf{x})\,. \tag{24.12}$$

In addition we have the length constraint for the homogeneous vector $\mathbf{h}$

$$h(\mathbf{h}) = \frac{1}{2}(\mathbf{h}^\mathsf{T}\mathbf{h} - 1) = 0\,. \tag{24.13}$$

The total differential of the $\boldsymbol{g}_i$ w. r. t. $\boldsymbol{q}_i$ and $\mathbf{h}$ is given by

$$A_i^\mathsf{T}\mathrm{d}\mathbf{h} + B_i^\mathsf{T}\mathrm{d}\mathbf{p}_i = \mathbf{0} \tag{24.14}$$

with the two Jacobians

$$A_i^\mathsf{T} = \frac{\partial \boldsymbol{g}_i}{\partial \mathbf{h}} = \underbrace{(\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}^{(l)}(\mathbf{q}_i)}_{2\times 9} \qquad B_i^\mathsf{T} = \frac{\partial \boldsymbol{g}_i}{\partial \mathbf{q}_i} = \underbrace{-\mathsf{S}^{(l)}(\mathsf{H}\mathbf{p}_i)}_{2\times 2} \tag{24.15}$$

Similarly, whe have the Jacobian of the length constraint

$$\frac{\partial h(\mathbf{h})}{\partial \mathbf{h}^\mathsf{T}} = \mathbf{h}^\mathsf{T}\,. \tag{24.16}$$

The Jacobians for all constraints $\boldsymbol{k}$ are given by

$$\underset{9\times 9}{A} = \begin{bmatrix} [A_i^\mathsf{T}] \\ \mathbf{h}^\mathsf{T} \end{bmatrix} = \begin{bmatrix} A_1^\mathsf{T} \\ A_2^\mathsf{T} \\ A_3^\mathsf{T} \\ A_4^\mathsf{T} \\ \mathbf{h}^\mathsf{T} \end{bmatrix} \tag{24.17}$$

$$\underset{9\times 12}{B^\mathsf{T}} = \begin{bmatrix} \mathrm{Diag}(\{B_i^\mathsf{T}\}) \\ \mathbf{0}^\mathsf{T} \end{bmatrix} = \begin{bmatrix} B_1^\mathsf{T} & 0^\mathsf{T} & 0^\mathsf{T} & 0^\mathsf{T} \\ 0^\mathsf{T} & B_2^\mathsf{T} & 0 & 0^\mathsf{T} \\ 0^\mathsf{T} & 0^\mathsf{T} & B_3^\mathsf{T} & 0^\mathsf{T} \\ 0^\mathsf{T} & 0^\mathsf{T} & 0^\mathsf{T} & B_4^\mathsf{T} \\ 0^\mathsf{T} & 0^\mathsf{T} & 0^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \tag{24.18}$$

If there are no singularities, the matrix A is regular, and we can determine random perturbation $\underline{\Delta\mathbf{h}}$ of the parameters $\mathbf{h}$ as a function of random perturbations $\underline{\Delta\boldsymbol{q}}$ of the observations $\mathbf{q}$ by

$$\underline{\Delta\mathbf{h}} = -A^{-1}B^\mathsf{T}\underline{\Delta\boldsymbol{q}} \tag{24.19}$$

We finally obtain the covariance matrix of $\mathbf{h}$

$$\Sigma_{\mathrm{hh}} = A^{-1}B^\mathsf{T}\Sigma_{xx}BA^{-\mathsf{T}}\,. \tag{24.20}$$

In this case the solution is unique.

Remark: (1) As to be expected the covariance matrix has rank 8, since the matrix B has rank 8. (2) We assumed the transformed points are given with their homogeneous coordinates. In case the homogeneous coordinates are Euclideanly normalized, thus $\mathbf{q}_i = [x_i, y_i, 1]^\mathsf{T}$, the third coordinate is fixed, thus has variance 0 and has correlations 0 with all other coordinates. Then the 12×12 covariance matrix Σ_{qq} is singular with rank 8, which is to be expected. However, we need not invert in our application. ◇

The complete algorithm for determining the homography from four point pairs and the uncertainty of the transformation parameters is given in the following. The two matrices M_1 and M_2 are used to condition the coordinates, thus centre and scale them.

Algorithm 4: Uncertain homography $\mathbf{q}_i = \mathsf{H}\mathbf{p}_i$ from four uncertain points
Input: $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$, $(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4)$, Σ_{qq}
Assume: $\Sigma_{pp} = \mathbf{0}$, $\det(\mathsf{H}) > 0$
Output: H, Σ_{hh}

for *all four points* **do**
  condition coordinates: $\mathbf{p}_i := \mathsf{M}_1\mathbf{p}_i$, $\mathbf{q}_i := \mathsf{M}_2\mathbf{q}_i, i = 1, 2, 3, 4$;
  coefficient matrices: $A_i = ([\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}^{(l)}(\mathbf{q}_i)])^\mathsf{T}$;
end
condition covariance matrix: $\Sigma_{xx} := (I_4 \otimes M_2)\Sigma_{xx})(I_4 \otimes M_2^\mathsf{T})$;
Design matrix: $A_0 := [A_i]$;
parameters: $\mathbf{h} = \mathsf{null}(A_0)$;
transformation: $\mathsf{H} := \mathrm{vec}^{-1}(\mathbf{h})$;
enforce determinant > 0: $s := \mathrm{sign}(\det(\mathsf{H}))$, $\mathbf{h} := s\mathbf{h}$, $\mathsf{H} := s\mathsf{H}$;
Jacobians of all constraints: $A = [A_0; \mathbf{h}^\mathsf{T}]$, $B^\mathsf{T} = -[\mathrm{Diag}(\mathsf{S}^{(l)}(\mathsf{H}\mathbf{p}_i)); \mathbf{0}^\mathsf{T}]$;
Jacobian od solution: $J := A^{-1}B^\mathsf{T}$;
covariance matrix of parameters: $\Sigma_{\mathrm{hh}} := J\Sigma_{xx}J^\mathsf{T}$;
uncondition transformation matrix: $\mathsf{H} := M_2^{-1}\mathsf{H}M_1$;
transformation of $\mathbf{h}$: $M := M_1^\mathsf{T} \otimes M_2^{-1}$;
uncondition covariance matrix: $\Sigma_{\mathrm{hh}} := M\Sigma_{\mathrm{hh}}M^\mathsf{T}$;

24.4 Both sets of coordinates are uncertain

We now have the complete oobservation vector with its, possibly full covariance matrix

$$\underset{24\times1}{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \quad \text{with} \quad \underset{24\times24}{\mathbb{D}(\boldsymbol{z})} = \Sigma_{zz}\,, \tag{24.21}$$

for the altogether 24 homogeneous coordinates $\boldsymbol{z}$ of the four point pairs.

We have the same constraints as above

$$\boldsymbol{g}_i(\mathbf{p}_i, \mathbf{q}_i, \mathbf{h}) = \mathsf{S}^{(l)}(\mathbf{q}_i) \;\; \mathsf{H} \;\; \mathbf{p}_i = -\mathsf{S}^{(l)}(\mathsf{H}\mathbf{p}_i)\,\mathbf{q}_i = (\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}^{(l)}(\mathbf{q}_i)\,\mathbf{h} = \mathbf{0} \quad \text{with} \quad i = 1, 2, 3, 4\,. \tag{24.22}$$

but now also depending on the observations $\boldsymbol{x} = [\mathbf{p}_i]$. The total differential is

$$A_i^\mathsf{T}\Delta\boldsymbol{h} + \underbrace{[B_{pi}^\mathsf{T} \mid B_{qi}^\mathsf{T}]}_{B_{pi}^\mathsf{T}} \begin{bmatrix} \Delta\boldsymbol{p}_i \\ \Delta\mathbf{q}_i \\ \Delta\boldsymbol{z} \end{bmatrix} = \mathbf{0} \tag{24.23}$$

with

$$\underset{2\times9}{A_i^\mathsf{T}} = \frac{\partial\boldsymbol{g}_i}{\partial\mathbf{h}} = \underbrace{(\mathbf{p}_i^\mathsf{T} \otimes \mathsf{S}^{(l)}(\mathbf{q}_i)}_{2\times9} \tag{24.24}$$

and

$$\underset{2\times3}{B_{pi}^\mathsf{T}} = \frac{\partial\boldsymbol{g}_i}{\partial\mathbf{p}_i} = \left(\mathsf{S}^{(l)}(\mathbf{q}_i)\mathsf{H}\right)^\mathsf{T} \qquad \underset{2\times3}{B_{qi}^\mathsf{T}} = \frac{\partial\boldsymbol{g}_i}{\partial\mathbf{q}_i} = -\mathsf{S}^{(l)}(\mathsf{H}\mathbf{p}_i)\,. \tag{24.25}$$

For all constraints we have

$$A\Delta\boldsymbol{h} + B\Delta\mathbf{z} = \mathbf{0} \tag{24.26}$$

with the matrices

$$\underset{9\times9}{A} = \begin{bmatrix} [A_i^\mathsf{T}] \\ \mathbf{h}^\mathsf{T} \end{bmatrix} = \begin{bmatrix} A_1^\mathsf{T} \\ A_2^\mathsf{T} \\ A_3^\mathsf{T} \\ A_4^\mathsf{T} \\ \mathbf{h}^\mathsf{T} \end{bmatrix} \tag{24.27}$$

and

$$\underset{9\times 24}{B^{\mathsf{T}}} = \begin{bmatrix} B_p^{\mathsf{T}} & B_q^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \end{bmatrix} := \begin{bmatrix} \mathrm{Diag}(\{B_{pi}^{\mathsf{T}}\}) & \mathrm{Diag}(\{B_{qi}^{\mathsf{T}}\}) \\ \mathbf{0}^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \end{bmatrix} \tag{24.28}$$

$$= \begin{bmatrix} B_1^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_1^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} \\ 0^{\mathsf{T}} & B_2^{\mathsf{T}} & 0 & 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_2^{\mathsf{T}} & 0 & 0^{\mathsf{T}} \\ 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_3^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_3^{\mathsf{T}} & 0^{\mathsf{T}} \\ 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_4^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & B_4^{\mathsf{T}} \\ 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & 0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \end{bmatrix} \tag{24.29}$$

similar to above.

We now can determine the covariance matrix. We discuss two cases.

1. As we in most cases can assume the $\mathbf{x}$ are independent of the $\mathbf{y}$ the covariance matrix of $\mathbf{h}$ reads as
$$\Sigma_{\mathrm{hh}} = A^{-1}B^{\mathsf{T}}\Sigma_{zz}BA^{-\mathsf{T}}. \tag{24.30}$$
2. Otherwise, we have more explicitely
$$\Sigma_{\mathrm{hh}} = A^{-1}[B_p^{\mathsf{T}} \mid B_q^{\mathsf{T}}] \begin{bmatrix} \Sigma_{\mathrm{pp}} & \Sigma_{\mathrm{pq}} \\ \Sigma_{\mathrm{qp}} & \Sigma_{\mathrm{qq}} \end{bmatrix} \begin{bmatrix} B_p \\ B_q \end{bmatrix} A^{-\mathsf{T}}. \tag{24.31}$$
Observe, the covariance matrices here refer to the homogeneous coordinates.
This might be relevant, if the corresponding points $(\boldsymbol{p}_i, \boldsymbol{q}_i)$, now in non-homogeneous coordinates, are determined using a key point detector in one image, leading to $\boldsymbol{p}_i$ and an intensity based method for measuring the parallaxes $\boldsymbol{d}_i = \boldsymbol{q}_i - \boldsymbol{p}_i$ e.g., using the Lucas-Kanade method or Least squares matching. Then the 16×16 covariance matrix of the 16 non-homogeneous coordinates of the four point pairs has the structure
$$\begin{bmatrix} \Sigma_{pp} & \Sigma_{pq} \\ \Sigma_{qp} & \Sigma_{qq} \end{bmatrix} = \begin{bmatrix} \mathrm{Diag}(\Sigma_{p_ip_i}) & \mathrm{Diag}(\Sigma_{p_iq_i}) \\ \mathrm{Diag}(\Sigma_{q_ip_i}) & \mathrm{Diag}(\Sigma_{q_iq_i}) \end{bmatrix} \tag{24.32}$$
with the covariance matrices $\Sigma_{p_ip_i}$ of the keypoints. Due to
$$\begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{q}_i \end{bmatrix} = \begin{bmatrix} I_2 & 0_2 \\ I_2 & I_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{d}_i \end{bmatrix} \tag{24.33}$$
we have
$$\Sigma_{p_iq_i} = \Sigma_{q_ip_i} = \Sigma_{p_ip_i} \tag{24.34}$$
and
$$\Sigma_{q_iq_i} = \Sigma_{p_ip_i} + \Sigma_{d_id_i}. \tag{24.35}$$

25 Uncertainty of Areas and Volumes

We give an explicit expression for the standard deviation of the area of a polygonal region determined by n points of homogeneous uncertainty. For regularly spaced points it essentially depends on the length of the circumference of the polygon.The result is generalized to the standard deviation of volumes of spatial regions, which essentially depend on the surface of the region.

25.1 Preface

The note (1999) was the written in the context of a lecture on Uncertainty in GIS. It has been extended to the uncertainty of surfaces of 3D regions.

25.2 The Problem

Areas and volumes are basic features of planar and spatial regions. Their accuracy often is required for subsequent tasks. This note gives simple expressions for the standard deviation of closed polygons and closed spatial regions.

25.3 Accuracy of the area of a polygon

25.3.1 Area of a polygon

The are of a polygon given by n points $\boldsymbol{p}_i(x_i, y_i), i = 1, ..., n$ can be determined via

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i + x_{i+1})(y_{i+1} - y_i) \tag{25.1}$$

$$= \frac{1}{2}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_{i+1} - x_i) \tag{25.2}$$

$$= \frac{1}{2}\sum_{i=0}^{n-1}x_i(y_{i+1} - y_{i-1}) \tag{25.3}$$

$$= \frac{1}{2}\sum_{i=0}^{n-1}y_i(x_{i+1} - x_{i-1})\,, \tag{25.4}$$

where the indices are to be taken modulo n, i.e., setting $(x_0, y_0) := (x_n, y_n)$ and $(x_{-1}, y_{-1}) := (x_{n-1}, y_{n-1})$.

Eq (25.2) results from (25.1) by exchanging x and y. Eq. (25.3) results from (25.1) by collecting corresponding terms

$$\begin{aligned} \sum (x_i + x_{i+1})(y_{i+1} - y_i) &= \sum x_i(y_{i+1} - y_i) + \sum x_{i+1}(y_{i+1} - y_i) && (25.5) \\ &= \sum x_i(y_{i+1} - y_i) + \sum x_i(y_i - y_{i-1}) && (25.6) \end{aligned}$$

Eq. (25.4) again results from (25.3) by exchanging x and y.

25.3.2 Accuracy of the area

We assume all coordinates have the same standard deviation $\sigma_{x_i} = \sigma_{y_i} = \sigma$ and are mutually independent.

The partial derivatives of the area w.r.t. coordinates are, using (25.3) and (25.4)

$$\begin{aligned} \frac{\partial A}{\partial x_i} &= \frac{1}{2}(y_{i+1} - y_{i-1}) && (25.7) \\ \frac{\partial A}{\partial y_i} &= \frac{1}{2}(x_{i+1} - x_{i-1}) && (25.8) \end{aligned}$$

Via variance propagation, therefore, we first obtain

$$\sigma_A^2 = \left[\frac{1}{4}\sum (x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2\right] \sigma^2 \tag{25.9}$$

With the length d_i of the i-th cord $(\boldsymbol{p}_{i-1}\boldsymbol{p}_{i+1})$ of the polygon we obtain

$$\sigma_A^2 = \frac{1}{4}\sum_{i=0}^{n-1} d_i^2 \, \sigma^2 \tag{25.10}$$

The standard deviation of the area therefore is

$$\sigma_A = \frac{1}{2}\sqrt{\sum_{i=0}^{n-1} d_i^2} \, \sigma \, . \tag{25.11}$$

This equation is rigorous for arbitrary polygons.

25.3.3 Approximation for densely sampled boundaries

In case the points of the polygon are equally spaced and sufficiently dense, e.g., when digitizing curved boundaries or when analysing (not too small) regions in raster images, we may approximate the length of the cord. With the length C of the circumference and the number of points n we assume

$$d_i = 2\frac{C}{n} \tag{25.12}$$

Then (25.10) simplifies to

$$\sigma_A^2 = \frac{C^2}{n}\sigma^2 \tag{25.13}$$

Thus, the standard deviation of the area is approximately

$$\sigma_A = \frac{C}{\sqrt{n}} \, \sigma \, . \tag{25.14}$$

As can be seen, the standard deviation of the area depends on

1. the length C of the region,
2. the number n of points, and
3. the standard deviation σ of the coordinates.

see Fig. . Especially, it does *not* depend on the area of the region.

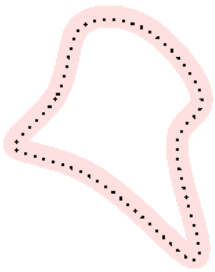


Figure 25.1: The uncertainty of the area of a region, whose boundary is densely sampled, dependes on the uncertainty of the points and on the circumference of the region, not its area

25.3.4 Accuracy of a rectangle

The expressions simplify for a rectangle. For the four points we always have the same chord length $d_i = d$, being the diagonal of the rectangle. Hence the standard deviation is given by aus Gl. (25.10)

$$\sigma_A = d\,\sigma \tag{25.15}$$

The standard deviation of the area of a rectangle thus only depends, among the uncertainty σ of the coordinates, from the length d of the diagonal.

25.4 Generalization to Volumes

We follow the argumentation of (25.10) to (25.14). We partition the surface into n surface elements s_i, of approximately the same size, and assume an uncertain point $\boldsymbol{p}_i$ in each of them. This may be visualized by triangulation, where around (nearly) each point we find a hexagonal region. Using the volume established by this surface element s_i and a change of the point $\boldsymbol{p}_i$ along the normal on the surface by Δn_i the spatial element, which has the form of a pyramid, has volume

$$\Delta V = \frac{1}{3} s_i \Delta n_i \,. \tag{25.16}$$

If we no assume homogeneous uncertainty of the surface points, i.e., assume $\underline{\Delta n}_i \sim \mathcal{M}(0, \sigma^2)$, then the i-th volume element has uncertainty

$$\sigma_i^2 = \left(\frac{s_i}{3}\right)^2 \sigma^2 \,. \tag{25.17}$$

Therefore, the variance of the complete volume is,

$$\sigma_V^2 = \frac{1}{9} \sum_i s_i^2 \sigma^2 \,, \tag{25.18}$$

see (25.10), replacing d_i by s_i. Hence, similar to the 2D case, where the elements d_i cover double the complete boundary, now the elements s_i cover three times the complete surface, if the surface is triangulated. Again, we assume the surface elements have similar size, – on an average, i.e., for large n, a hexagon –

$$s_i = \frac{3S}{n} \,, \tag{25.19}$$

since the number of hexagons is approximately identical to the number of points and cover each tringle three times.

Then the variance simplifies to

$$\sigma_V^2 = \frac{1}{9}\left(\frac{3S^2}{n^2}\right)\left(\sum_{i=1}^{n} 1\right)\sigma^2 = \frac{S^2}{n^2}\, n\sigma^2 = \frac{S^2}{n}\,\sigma^2\,. \tag{25.20}$$

This, finally, leads to the standard deviation of the volume V

$$\sigma_V = \frac{S}{\sqrt{n}}\sigma\,. \tag{25.21}$$

As to be expected: the standard deviation of the volume of a spatial region, given by surface points of homogeneous uncertainty, increases with the surface of the region.

Part IV

Technical Notes on Image Analysis

26 Checking of Least Squares Matching Using Image Triplets

We discuss how to statistically check the accuracy of least squares matching, which estimates affine geometric and radiometric transformations of image pairs, using the circular closure of the three transformations in an image triplet.

26.1 Preface

The note (2014) together with the following notes addresses the task of least squares matching. We start with constraints for checking LSM results using image triples, by requiring the product of the geometric and the radiometric transformations are the unit transformation.

26.2 Goal

Let the pairwise geometric and radiometric transformations between three images be captured by $\{\boldsymbol{\theta}_i, \Sigma_{ii}\}, i \in [1,2,3]$ with the partitioning

$$\boldsymbol{\theta}_i = \begin{bmatrix} \boldsymbol{\theta}_G \\ \boldsymbol{\theta}_R \end{bmatrix}_i = \begin{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} \\ \begin{bmatrix} \theta_7 \\ \theta_8 \end{bmatrix} \end{bmatrix}_i . \tag{26.1}$$

The index indicates an image pair $(i, i+1)$, thus stands for the first of one of the three images, and assumes the indices are taken cyclically, see Fig. 26.1.

The geometric and the radiometric transformations are given by the homogeneous matrices

$$\mathsf{G}_i(\boldsymbol{\theta}_{G_i}) = \begin{bmatrix} A & \boldsymbol{p} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}_i = \begin{bmatrix} \theta_1 & \theta_3 & \theta_5 \\ \theta_2 & \theta_4 & \theta_6 \\ 0 & 0 & 1 \end{bmatrix}_i \tag{26.2}$$

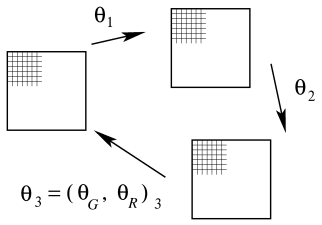


Figure 26.1: Checking matching results using image triplets

and

$$\mathsf{R}_i(\boldsymbol{\theta}_{R_i}) = \begin{bmatrix} s & t \\ 0 & 1 \end{bmatrix}_i = \begin{bmatrix} \theta_7 & \theta_8 \\ 0 & 1 \end{bmatrix}_i \tag{26.3}$$

referring to the multiplicative parts A and s and the additive parts $\boldsymbol{p}$ and t.[1]

The covariance matrices refers to additive corrections, i.e., to the corrections $\Delta\boldsymbol{\theta}_i$:

$$\underline{\mathsf{G}} = \mathsf{G}^a + \underline{\Delta\mathsf{G}} \quad \text{and} \quad \underline{\mathsf{R}} = \mathsf{R}^a + \underline{\Delta\mathsf{R}} \tag{26.4}$$

with

$$\Delta\mathsf{G} = \begin{bmatrix} \Delta\theta_1 & \Delta\theta_3 & \Delta\theta_5 \\ \Delta\theta_2 & \Delta\theta_4 & \Delta\theta_6 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \Delta\mathsf{R} = \begin{bmatrix} \Delta\theta_1 & \Delta\theta_2 \\ 0 & 0 \end{bmatrix}. \tag{26.5}$$

The **task** is to check the consistency using the constraints

$$\prod_{i=1}^{3} \mathsf{G}_i = I_3 \quad \text{and} \quad \prod_{i=1}^{3} \mathsf{R}_i = I_2\,. \tag{26.6}$$

26.3 Setup

Since the parameters of the geometric and the radiometric transformations are correlated we concatenate the two transformations to

$$\mathsf{M}(\boldsymbol{\theta}_i) = \begin{bmatrix} \mathsf{G}(\boldsymbol{\theta}_{Gi}) & 0 \\ 0 & \mathsf{R}(\boldsymbol{\theta}_{Ri}) \end{bmatrix} \tag{26.7}$$

and check the constraint

$$\boldsymbol{\theta}_1 \circ \boldsymbol{\theta}_2 \circ \boldsymbol{\theta}_3 = \mathbf{1}: \quad \prod_{i=1}^{3} \mathsf{M}_i = I_5 \quad \text{with} \quad \mathbf{1} = \begin{bmatrix} \boldsymbol{e}_1^{[6]} \\ \boldsymbol{e}_1^{[2]} \end{bmatrix}. \tag{26.8}$$

Observe, due to the special structure this constraint only contains 8 individual constraints depending on the 24 parameters of the three transformations.

26.3.1 The radiometric constraint

We collect all given parameters in the 24-vector

$$\boldsymbol{\theta} = [\theta_u]_{u=1,\ldots,24} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_3 \end{bmatrix}. \tag{26.9}$$

Using the transformation matrices, the radiometric constraint reads as

$$0 = \mathsf{R}_1\mathsf{R}_2\mathsf{R}_3 - I_2\,. \tag{26.10}$$

[1] Observe, the matrix R in this note refers to the radiometric transformation between two images and not to a rotation matrix.

The two linearized constraints, see (26.5), right, for the elements of the first row are:

$$\mathbf{0} = \boldsymbol{c}_R = \begin{bmatrix} \theta_7\,\theta_{15}\,\theta_{23} - 1 \\ \theta_8 + \theta_7\,\theta_{16} + \theta_7\,\theta_{15}\,\theta_{24} \end{bmatrix} \tag{26.11}$$

The Jacobian w.r.t. the 6 parameters $\boldsymbol{\theta}_R$ are

$$\underset{6\times 2}{J_R^{\mathsf{T}}} = \begin{bmatrix} \underset{2\times 2}{J_{R_1}^{\mathsf{T}}} \\ \underset{2\times 2}{J_{R_2}^{\mathsf{T}}} \\ \underset{2\times 2}{J_{R_3}^{\mathsf{T}}} \end{bmatrix} = \left[\frac{\partial \boldsymbol{c}_R}{\partial \boldsymbol{\theta}_R}\right]^{\mathsf{T}} = [J_s\, J_t] \begin{bmatrix} \theta_{15}\,\theta_{23} & \theta_{16} + \theta_{15}\,\theta_{24} \\ 0 & 1 \\ \hline\hline \theta_7\,\theta_{23} & \theta_7\,\theta_{24} \\ 0 & \theta_7 \\ \hline\hline \theta_7\,\theta_{15} & 0 \\ 0 & \theta_7\,\theta_{15} \end{bmatrix} \tag{26.12}$$

26.3.2 The geometric constraint

The geometric constraint reads as

$$0 = \mathsf{G}_1\mathsf{G}_2\mathsf{G}_3 - I_3\,. \tag{26.13}$$

The six linearized constraints, see (26.5), left, for the elements of the first two rows are:

$$\mathbf{0} = \boldsymbol{c}_G = \begin{bmatrix} \theta_{17}\ (\theta_1\,\theta_9 + \theta_3\,\theta_{10}) + \theta_{18}\ (\theta_1\,\theta_{11} + \theta_3\,\theta_{12}) \\ \theta_{17}\ (\theta_2\,\theta_9 + \theta_4\,\theta_{10}) + \theta_{18}\ (\theta_2\,\theta_{11} + \theta_4\,\theta_{12}) \\ \theta_{19}\ (\theta_1\,\theta_9 + \theta_3\,\theta_{10}) + \theta_{20}\ (\theta_1\,\theta_{11} + \theta_3\,\theta_{12}) \\ \theta_{19}\ (\theta_2\,\theta_9 + \theta_4\,\theta_{10}) + \theta_{20}\ (\theta_2\,\theta_{11} + \theta_4\,\theta_{12}) \\ \hline \theta_5 + \theta_1\,\theta_{13} + \theta_3\,\theta_{14} + \theta_{21}\ (\theta_1\,\theta_9 + \theta_3\,\theta_{10}) + \theta_{22}\ (\theta_1\,\theta_{11} + \theta_3\,\theta_{12}) \\ \theta_6 + \theta_2\,\theta_{13} + \theta_4\,\theta_{14} + \theta_{21}\ (\theta_2\,\theta_9 + \theta_4\,\theta_{10}) + \theta_{22}\ (\theta_2\,\theta_{11} + \theta_4\,\theta_{12}) \end{bmatrix} \tag{26.14}$$

The transposed Jacobian is

$$\underset{18\times 6}{J_G^{\mathsf{T}}} = \begin{bmatrix} \underset{6\times 6}{J_{G_1}^{\mathsf{T}}} \\ \underset{6\times 6}{J_{G_2}^{\mathsf{T}}} \\ \underset{6\times 6}{J_{G_3}^{\mathsf{T}}} \end{bmatrix} = \left[\frac{\partial \boldsymbol{c}_G}{\partial \boldsymbol{\theta}_G}\right]^{\mathsf{T}} = [J_a^{\mathsf{T}}, J_p^{\mathsf{T}}] \tag{26.15}$$

with

$$J_a^{\mathsf{T}} = \begin{bmatrix} \theta_9\,\theta_{17} + \theta_{11}\,\theta_{18} & 0 & \theta_9\,\theta_{19} + \theta_{11}\,\theta_{20} & 0 \\ 0 & \theta_9\,\theta_{17} + \theta_{11}\,\theta_{18} & 0 & \theta_9\,\theta_{19} + \theta_{11}\,\theta_{20} \\ \theta_{10}\,\theta_{17} + \theta_{12}\,\theta_{18} & 0 & \theta_{10}\,\theta_{19} + \theta_{12}\,\theta_{20} & 0 \\ 0 & \theta_{10}\,\theta_{17} + \theta_{12}\,\theta_{18} & 0 & \theta_{10}\,\theta_{19} + \theta_{12}\,\theta_{20} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline\hline \theta_1\,\theta_{17} & \theta_2\,\theta_{17} & \theta_1\,\theta_{19} & \theta_2\,\theta_{19} \\ \theta_3\,\theta_{17} & \theta_4\,\theta_{17} & \theta_3\,\theta_{19} & \theta_4\,\theta_{19} \\ \theta_1\,\theta_{18} & \theta_2\,\theta_{18} & \theta_1\,\theta_{20} & \theta_2\,\theta_{20} \\ \theta_3\,\theta_{18} & \theta_4\,\theta_{18} & \theta_3\,\theta_{20} & \theta_4\,\theta_{20} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline\hline \theta_1\,\theta_9 + \theta_3\,\theta_{10} & \theta_2\,\theta_9 + \theta_4\,\theta_{10} & 0 & 0 \\ \theta_1\,\theta_{11} + \theta_3\,\theta_{12} & \theta_2\,\theta_{11} + \theta_4\,\theta_{12} & 0 & 0 \\ 0 & 0 & \theta_1\,\theta_9 + \theta_3\,\theta_{10} & \theta_2\,\theta_9 + \theta_4\,\theta_{10} \\ 0 & 0 & \theta_1\,\theta_{11} + \theta_3\,\theta_{12} & \theta_2\,\theta_{11} + \theta_4\,\theta_{12} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{26.16}$$

and

$$J_p^\mathsf{T} = \left[\begin{array}{cc}
\theta_{13} + \theta_9\,\theta_{21} + \theta_{11}\,\theta_{22} & 0 \\
0 & \theta_{13} + \theta_9\,\theta_{21} + \theta_{11}\,\theta_{22} \\
\theta_{14} + \theta_{10}\,\theta_{21} + \theta_{12}\,\theta_{22} & 0 \\
0 & \theta_{14} + \theta_{10}\,\theta_{21} + \theta_{12}\,\theta_{22} \\
\hline
1 & 0 \\
0 & 1 \\
\hline\hline
\theta_1\,\theta_{21} & \theta_2\,\theta_{21} \\
\theta_3\,\theta_{21} & \theta_4\,\theta_{21} \\
\theta_1\,\theta_{22} & \theta_2\,\theta_{22} \\
\theta_3\,\theta_{22} & \theta_4\,\theta_{22} \\
\hline
\theta_1 & \theta_2 \\
\theta_3 & \theta_4 \\
\hline\hline
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\hline
\theta_1\,\theta_9 + \theta_3\,\theta_{10} & \theta_2\,\theta_9 + \theta_4\,\theta_{10} \\
\theta_1\,\theta_{11} + \theta_3\,\theta_{12} & \theta_2\,\theta_{11} + \theta_4\,\theta_{12}
\end{array}\right] \tag{26.17}$$

Observe the equivalence in the structure of the two Jacobians for the radiometric and the geometric constraints: scalar values in J_R correspond to 2×2 and 4×2 matrices in J_G.

26.4 The joint constraint and the statistical test

26.4.1 The constraint

The joint constraint for the geometric and the radiometric parameters is

$$\mathbf{0} = \boldsymbol{c} = \left[\begin{array}{c} \boldsymbol{c}_g \\ \boldsymbol{c}_r \end{array}\right] \tag{26.18}$$

Its Jacobian is

$$J_{c\theta} = \underset{8\times 24}{\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\theta}}} = \left[\begin{array}{cccccc} J_{G_1} & 0 & J_{G_2} & 0 & J_{G_3} & 0 \\ 0 & J_{R_1} & 0 & J_{R_2} & 0 & J_{R_3} \end{array}\right]. \tag{26.19}$$

26.4.2 The covariance matrix of the parameters

The covariance matrices refer to the transformations, which themselves depend on the used intensities. Since the same intensities are used in neighbouring transformations the resultant parameter vectors are not uncorrelated. The correlation depends on the number of common pixels used and – at least – on the gradients at these pixels. A rigorous derivation of the correlations appears prohibitive.

We therefore derive a correlation structure which reflects the mutual dependencies assuming the transformations themselves are close to a unit transformation. We derive the correlation structure using a substitute problem.

Let the three intensities in the three images at a certain position be $\boldsymbol{g} = [g_i], i = 1, 2, 3$. Then the three intensity differences correspond to the transformations and are

$$\left[\begin{array}{c} \Delta g_1 \\ \Delta g_2 \\ \Delta g_3 \end{array}\right] := \left[\begin{array}{c} \Delta g_{12} \\ \Delta g_{23} \\ \Delta g_{31} \end{array}\right] = \left[\begin{array}{c} g_2 - g_1 \\ g_3 - g_2 \\ g_1 - g_3 \end{array}\right] = D\boldsymbol{g} \tag{26.20}$$

with

$$D = \left[\begin{array}{ccc} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{array}\right]. \tag{26.21}$$

Hence, if the covariance matrix of the intensities is $\mathbb{D}(\boldsymbol{g}) = \sigma_g I_3$, then the differences have the covariance matrix

$$\Sigma_{\Delta g \Delta g} = \sigma_g^2 D D^\mathsf{T} = \sigma_g^2 \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \tag{26.22}$$

which is singular, since $\Sigma_{\Delta g \Delta g} \mathbf{1} = \mathbf{0}$. The correlation matrix is

$$\mathrm{Corr}(\Delta g) = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{26.23}$$

We use this correlation structure and transfer it to the covariance matrix of all parameters $\boldsymbol{\theta}$. Let the matrix square roots of the three covariance matrices be

$$S_i = \sqrt{\Sigma_i}\,. \tag{26.24}$$

Then we use as covariance matrix for the parameters

$$\Sigma_{\theta\theta} = \frac{1}{2} \begin{bmatrix} 2\Sigma_1 & -S_1 S_2 & -S_1 S_3 \\ -S_2 S_1 & 2\Sigma_2 & -S_2 S_3 \\ -S_3 S_1 & -S_3 S_2 & 2\Sigma_3 \end{bmatrix}. \tag{26.25}$$

26.4.3 The statistical test

The covariance matrix of the constraints now is

$$\Sigma_{cc} = J_{c\theta} \Sigma_{\theta\theta} J_{c\theta}^\mathsf{T} \tag{26.26}$$

The test statistics therefore is

$$\underline{X} = \underline{\boldsymbol{c}}^\mathsf{T} \Sigma_{cc}^{-1} \underline{\boldsymbol{c}} \sim \chi_8^2\,. \tag{26.27}$$

27 Asymmetric and Symmetric Matching with experiments

The note addresses template matching and least squares matching of one- and two-dimensional signals useful for high precision image matching. We derive the method for linear geometric and radiometric transformations and for an asymmetric and a symmetric setup. The goal is to arrive at methods which provide reliable results not only for the estimated transformation parameters, but also for their covariance matrix and the estimated variance factor. This requires to also take the effects of interpolation and smoothing into account. We provide preliminary experimental results.

27.1 Summary

The report addresses least squares matching (LSM) of one- and two-dimensional scalar valued signals useful for high precision image matching.

The main results are the following:

1. LSM can be realized such that – for sets of simulated data – the three tests on the correctness of the implementation do not fire:

(a) The estimated variance factor does significantly differ from 1. This indicates that model and (simulated) data are consistent.

(b) The empirical covariance matrix derived from samples does not significantly differ from the theoretical covariance matrix, which is the Cramer-Rao bound. This suggests, that the theoretical covariance matrix can be used as reliable uncertainty indicator.

(c) The estimates show no significant bias.

2. LSM can be realized symmetrically. Then exchanging the two signals leads to identical results, i.e., mutually inverse geometric and radiometric transformations.

3. Individual variances for all observations can be taken into account, especially, position or signal depending variances. Covariances are neglected.

4. Interpolation of the observed signals may be necessary:

   - Template matching, where one of the two signals is known as continuous function, does not require any interpolation of observed data.
   - Asymmetric LSM requires interpolation of one of the two observed signals. The effect of interpolation onto the noise properties can be predicted and thus taken into account. This especially holds for interpolation schemes where the interpolation is a linear function of the given signal, such as linear or cubic interpolation.
   - Symmetric LSM of two observed signals partially requires interpolation of observed data:
     - The observation equations are built on the observed data, thus no interpolation is necessary..
     - The estimation of the underlying true signal requires interpolation/smoothing of the observed data.

5. For affine geometric transformations the similarity transformation derivable from corresponding Lowe-keypoints can be used as approximate transformation. This allows to derive affine matches with a covariance matrix of the parameters of the affinity, which can be used for estimating the relative pose of two calibrated or partially calibrated cameras, i.e., with or without focal length based on pairs of affine matches.

27.2 Goals

This report addresses least-squares matching (LSM) of one- and two-dimensional scalar valued signals. Given are two signals g and h and a parametric transformation for geometry and intensity: The task is to optimally determine the parameters and their quality.

The following observations indicate that this task is challenging:

- Exchanging g and h may not lead to the same result, i.e., the inverse transformation. Often, exchanging the two signals is used to check the quality of the match. However, there should be a solution which is invariant to the exchange of the two signals.
- The predicted quality, namely the variance or the covariance matrix, often is far too optimistic. Though the Cramer-Rao bound, derivable from the estimation process, is known to be a lower bound for the variance, at least when using simulated data which fulfil the estimation model, the bound should be reachable.
- The noise characteristics of the two signals often significantly deviates from the assumptions made in the estimation model. This may be caused by simply and erroneously assuming the given signal values are independent and identically distributed (i.i.d.). Or it may result from interpolation effects. Therefore, among other indicators, the estimated variance factor significantly deviates from 1. Again, at least if

the signals are simulated, these effects should be tractable and lead to an unbiased estimate for the variance factor.

- The two signals can be seen as a discrete noisy versions of an unknown continuous underlying true signal f. This leads to two problems which may hinder an optimal solution:
  - the discrete values do not refer to the same position of f. This requires adequate interpolation, hence, implicitly relies on some model for f.
  - the method needs to estimate f together with the geometric and intensity parameters.

  Depending on the realization, this may lead to biases in the parameters and the estimated variance factor.
- The computational complexity mainly depends on the size of the image patches. Even with a hierarchical approach the computation time may not be acceptable for a specific (time critical) application.

The report addresses these aspects and tries to answer the following questions:

1. Does template matching, where one of the two signals is perfectly known, show the above mentioned effects w.r.t. bias of estimated variance, variance of estimated parameters, and estimated parameters?
2. Is there a symmetric solution, which is invariant to the exchange of the two signals? How far do the results differ from an asymmetric solution?
3. What is the effect of possible interpolation schemes?
4. How should the unknown signal f be estimated and how does this choice effect the quality of the result?
5. How could the computational complexity of the estimation process be reduced?

On Notation. Signals are one- or two-dimensional function. The function names taken from the middle of the alphabet, e.g., f, g; and h. Coordinate names are taken from the end of the alphabet: so e.g., $f(\boldsymbol{x})$, $g(\boldsymbol{y})$, and $h(\boldsymbol{z})$. Sometimes, we use the convention $\boldsymbol{x} = [x, y]^\mathsf{T}$. Discrete functions depend on coordinates x_i or $\boldsymbol{x}_i$, where the index range is a set of integers $\in \mathbb{Z}$ With the grid spacing Δx we generally have $x_i = i\Delta x$; mostly, we assume the spacing $\Delta x = 1$, hence we simply have $x_i = i$. If a coordinate x has no index, the context tells whether it is a real or an integer. If two coordinates, say $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ have the same index they refer to corresponding points. Homogeneous coordinates and matrices are boldface upright, e.g., coordinates $\mathbf{x}$ or transformation A. The dimension depends on the context. Stochastical variables are underscored, e.g., the discrete noise function is $\underline{m}(\boldsymbol{x}_i)$.

27.3 Image Matching

27.3.1 Models

We first discuss template matching and asymmetric and symmetric image matching, without specifying the meaning of the transformation parameters. Then we will discuss several transformation and noise models.

27.3.1.1 Template matching

We assume a template $f(\boldsymbol{x})$ is given as a continuous function. The observed signal h is a geometrically and radiometrically distorted and noisy version of the template, specified

by the geometric parameters $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_R$ and parameters $\boldsymbol{\theta}_n$ characterizing the noise. The intensity transformation is

$$f \mapsto h: \quad h = \mathcal{T}_R(f, \boldsymbol{\theta}_R)\,. \tag{27.1}$$

The geometric transformation is

$$\boldsymbol{x} \mapsto \boldsymbol{z}: \quad \boldsymbol{z} = \mathcal{T}_G(\boldsymbol{x}, \boldsymbol{\theta}_G)\,. \tag{27.2}$$

Hence we have the observed signal

$$\underline{h}(\boldsymbol{z}_i) = h(\boldsymbol{z}_i) + \underline{m}(\boldsymbol{z}_i, \boldsymbol{\theta}_n)\,, \quad i = 1, ..., I\,, \tag{27.3}$$

or compound:

$$h(\boldsymbol{z}_i) = \mathcal{T}_R(f(\mathcal{T}_G^{-1}(\boldsymbol{z}_i, \boldsymbol{\theta}_G)), \boldsymbol{\theta}_R) + m(\boldsymbol{z}_i, \boldsymbol{\theta}_n)\,, \quad i = 1, ..., I\,, \tag{27.4}$$

assuming the noise components are uncorrelated, i.e., $\mathbb{C}\mathrm{ov}(\underline{m}_i, \underline{m}_j) = \sigma_{ij}\delta_{i,j}$.

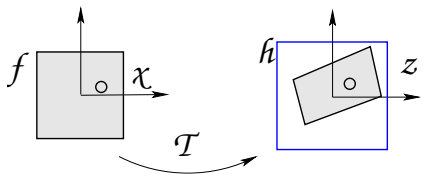


Figure 27.1: Model of template matching: given template f, observed signal g, intensity transformation $\mathcal{T}_R : f \mapsto h$, geometric transformation $\mathcal{T}_G : \boldsymbol{x} \mapsto \boldsymbol{z}$

Example 27.3.29: ***Template matching with geometric shift and radiometric affinity.*** Let the geometric transformation be a translation

$$\mathcal{T}_G: \quad \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{u} \tag{27.5}$$

and the radiometric transformation be an affinity, compensating for brightness and contrast

$$\mathcal{T}_R: \quad h = s\, f + t \tag{27.6}$$

then we have the compound model

$$\underline{h}(\boldsymbol{z}_i) = s\, f(\boldsymbol{z}_i - \boldsymbol{u}) + t + \underline{m}(\boldsymbol{z}_i)\,, \qquad i = 1, ..., I\,. \tag{27.7}$$

This model also is assumed when determining the shift $\boldsymbol{u}$ via the normalized cross-correlation:

$$\widehat{\boldsymbol{u}} = \mathrm{argmax}_u\, \rho(f(\boldsymbol{x}), h(\boldsymbol{z} - \boldsymbol{u})) = \mathrm{argmax}_u \frac{\mathbb{C}\mathrm{ov}(f(\boldsymbol{x}), h(\boldsymbol{z} - \boldsymbol{u}))}{\sqrt{\mathbb{V}(f(\boldsymbol{x}))\mathbb{V}(h(\boldsymbol{z} - \boldsymbol{u}))}}\,. \tag{27.8}$$

where the operators $\mathbb{C}\mathrm{ov}(.)$ and $\mathbb{V}(.)$ determine the empirical covariance and variance, which are invariant to (brightness) shifts. If the variances $\mathbb{V}(h(\boldsymbol{z} - \boldsymbol{u}))$ are independent of $\boldsymbol{u}$, then we find that $\boldsymbol{u}$ which maximizes the un-normalized correlation

$$\boldsymbol{u} = \mathrm{argmax}_u\, c(\boldsymbol{u}) = \mathrm{argmax}_u\, f(\boldsymbol{u}) * h(-\boldsymbol{u})\,, \tag{27.9}$$

where '$*$' denotes convolution. ◇

Observe the following special properties and limitations motivating image matching:

- Whereas the radiometric mapping $\mathcal{T}_R$ is a forward mapping, from the given template to the observable signal, the geometric mapping $\mathcal{T}_G$ is used as backward mapping, corresponding to using the inverse transformation in image warping.
- We assume f is continuous, though in reality we will have a discrete sequence $f(\boldsymbol{x}_i)$ and some interpolation scheme. We will require, that f is C^1 continuous, i.e., the function has unique first derivatives.

- The number I of observations depends on whether the function is known in $\mathbb{R}^2$, then all pixels of h can be used, or only in a limited region, as depicted in the figure. Then, only those pixels in h can be used, which map into the region, where f is known. We discuss boundary effects later.
- The noise component refers to the observed signal $h(\boldsymbol{z}_i)$. Therefore, no interpolation of the noisy observed signal is necessary.
- One easily can criticize the above radiometric model. Physically brightness changes may have many causes, e.g., change in illumination. The resulting image is perceived as brighter, but actually the contrast is increased. Based on this view, the above meaning attributed to s and t appears to be not correct. This is the reason why Koenderink and Doorn (2002) proposed to model the logarithms of the intensities. Replacing the signals by their logarithm leads to the model (neglecting the spatial dependency of the brightness)

$$\mathcal{T}'_R: \quad \underline{\log h} = \gamma \log f + \zeta + \underline{\mu}' \tag{27.10}$$

or taking exponents on both sides

$$\mathcal{T}'_R: \quad \underline{h} = s\, f^{\gamma}\, \underline{m}' \quad \text{with} \quad s = \exp\zeta \quad \text{and} \quad m' = \exp\mu' \,. \tag{27.11}$$

The factor s can be related to the physical change of brightness[1] and the exponent γ represents what is known as γ correction, used for modelling the perception of the human eye. Of course, the statistics of the noise component then changes: it is multiplicative.
In the following, we leave open whether we choose the intensities f or the log-intensities $\log(f)$ for image matching and only derive the equations for the intensities.

27.3.1.2 The asymmetric image matching model

In case two images are given, we have the asymmetric model

$$\underline{g}(\boldsymbol{y}_i) = f(\boldsymbol{x}_i) + \underline{n}(\boldsymbol{y}_i)\,, \quad i = 1, ..., I \tag{27.12}$$

$$\underline{h}(\boldsymbol{z}_j) = \mathcal{T}_R(f(\mathcal{T}_G^{-1}(\boldsymbol{z}_j, \boldsymbol{\theta}_G)), \boldsymbol{\theta}_I) + \underline{m}(\boldsymbol{z}_j, \boldsymbol{\theta}_n)\,, \quad j = 1, ..., J\,. \tag{27.13}$$

The model is name asymmetric, since the radiometric and geometric transformations are only affecting the second image h. The number $N = I + J$ of observed intensities depends on the geometric transformation: If we assume that all pixels from g map into the region of h, as in the figure, then only those J pixels in h which map to the region of g are useful. Otherwise we need to identify the common overlap and also take only I pixels in g which map into the region of h and only the function f within the common regions needs to be estimated. Of course, only a rectangular region within g may be taken, as indicated by the dashed boundary in the right part of the Fig. 27.2.

Here we assume an identity transformation

$$\boldsymbol{y} = \boldsymbol{x} \tag{27.14}$$

between the coordinates of the unknown template $f(\boldsymbol{x})$ and the observable image $g(\boldsymbol{y})$, the noise components $\underline{n}(\boldsymbol{y}_i)$ and $\underline{m}(\boldsymbol{z}_j)$, and the intensity transformation

$$h = \mathcal{T}_R(f, \boldsymbol{\theta}_I) \tag{27.15}$$

and the geometric transformation

$$\boldsymbol{z} = \mathcal{T}_G(\boldsymbol{y}, \boldsymbol{\theta}_G) \tag{27.16}$$

[1] Koenderink and Doorn (2002) model $\gamma = \exp\delta$ to ensure positivity of γ and obtaining an exponent $\gamma = 1$ for $\delta = 0$.

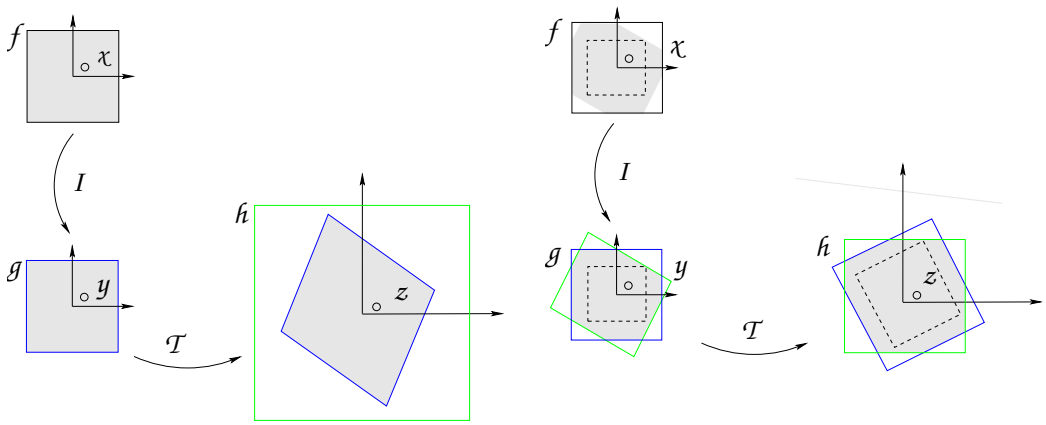


Figure 27.2: Model of asymmetric image matching: unknown template f, observed signals g and h, intensity transformation $\mathcal{T}_R : g \mapsto h$, geometric transformation $\mathcal{T}_G : \boldsymbol{y} \mapsto \boldsymbol{z}$. **Left:** region g is contained in region h. **Right:** The regions g and h overlap

with their unknown parameters $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_G$ respectively. In addition to the parameters $\boldsymbol{\theta}_T$, $\boldsymbol{\theta}_I$, and $\boldsymbol{\theta}_n$, also the function C^1-continuous f is unknown.

Observe, that the model only simplifies to template matching if the noise of $g(\boldsymbol{x}_i)$ is zero and the discrete values $f(\boldsymbol{x}_i) \equiv g(\boldsymbol{x}_i)$ rigorously allow to derive the continuous function $f(\boldsymbol{x})$, e.g., by proper interpolation. Otherwise we will encounter interpolation errors.

27.3.1.3 The symmetric model

Sometimes it appears more intuitive to make the model symmetric, as suggested by G. Vosselman.

Symmetric image matching assumes a transformation from the unknown template to each observed image, see Fig. 27.3.

We have three signals given in their own coordinate system

- The unknown signal $f(\boldsymbol{x})$. It can be treated as a stochastic signal, without specifying its stochastic properties at the moment.
- Two observed signals $g(\boldsymbol{y})$ and $h(\boldsymbol{z})$, whose true values $\tilde{g}$ and $\tilde{h}$ are related to the unknown signal by some geometric and intensity transformation.

The relation between these signals is the following

- The geometric model: We have correspondent positions

$$\big(\mathcal{X}(\boldsymbol{x}), y(\boldsymbol{y}), z(\boldsymbol{z})\big) \tag{27.17}$$

We thus have the geometric mappings

$$y \mapsto \mathcal{X}: \quad \mathcal{X} = \mathcal{A}_1(y) \tag{27.18}$$
$$\mathcal{X} \mapsto z: \quad z = \mathcal{A}_2(\mathcal{X}) \tag{27.19}$$

Thus the mapping between the two images is

$$y \mapsto z: \quad z = \mathcal{B}(y) \quad \text{with} \quad \mathcal{B} = \mathcal{A}_2 \circ \mathcal{A}_1 \tag{27.20}$$

Of course, only the parameters of the compound mapping are estimable. This can be achieved, by either setting $\mathcal{A}_1$ or $\mathcal{A}_2$ to the unit matrix, as in the previous section where $\mathcal{A}_1 = I$ or to impose a constraint between $\mathcal{A}_1$ and $\mathcal{A}_2$. A similar argument holds for the intensity mappings, hence, $\mathcal{T}_R = \mathcal{S}_2\mathcal{S}_1$.

In order to achieve symmetry of the transformations, we partition the intensity and geometry transformations into two parts, each performing half of the transformation. In order to keep the model simple, we explicitly model half of the transformations and then determine the combined transformation, being the product of these transformations, instead of modelling the total transformations and then needing to their square root. We assume the compound geometric mapping to be $\mathcal{T}_G = \mathcal{A}^2 := \mathcal{A} \circ \mathcal{A}$,

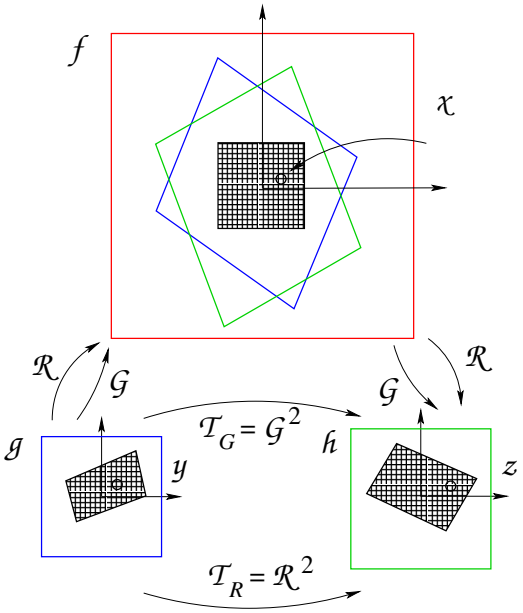


Figure 27.3: Relations between two given image patches $g : g(\boldsymbol{y})$ (blue) and $h : h(\boldsymbol{h})$ (green) and the mean patch $f : f(\boldsymbol{x})$ (which is the black within the red region). The two image patches g and h are related by a geometric (not named here) and a radiometric affinity $\mathcal{S}$, respectively. The correspondence is established by the patch f. Geometrically and radiometrically it lies in the middle between g and h. Only a region in the overlap of the two patches g and h mapped to f can be used. We choose the maximum square (black). The observations are all pixels in g and h which map into the black square of the reference image f. We assume the reference image $\hat{f}$ is a restored version of the weighted mean of the two projected images g and h

implicitly saying the coordinate system of x to be in the middle of the coordinate systems of y and z. This reads as

$$y \mapsto x: \quad x = \mathcal{A}(y, \boldsymbol{\theta}_G) \tag{27.21}$$
$$x \mapsto z: \quad z = \mathcal{A}(x, \boldsymbol{\theta}_G) \tag{27.22}$$

with the unknown geometric parameters $\boldsymbol{\theta}_G$.

- The radiometric model: The observed intensities g and h are given at discrete points

$$\{\{g(y_j)\}, j = 1, \ldots, J, \{h(z_j\}, k = 1, \ldots, K\} \tag{27.23}$$

not necessarily corresponding.
We again use a symmetric set-up leading to the relations valid at corresponding points

$$g \mapsto f: \quad f = \mathcal{R}(g, \boldsymbol{\theta}_I) \tag{27.24}$$
$$f \mapsto h: \quad h = \mathcal{R}(f, \boldsymbol{\theta}_I) \tag{27.25}$$

with the unknown intensity parameters $\boldsymbol{\theta}_I$.

Integrating the geometry and intensity transformation we arrive at the following model, which is generative, i.e., allows to simulate observed images

$$g(\boldsymbol{y}_j) = \mathcal{R}^{-1}\left(f\left(\mathcal{A}(\boldsymbol{y}_j, \boldsymbol{\theta}_G), \boldsymbol{\theta}_I\right) + \underline{n}(\boldsymbol{y}_j), \quad j = 1, \ldots, J \tag{27.26}$$

$$h(\boldsymbol{z}_k) = \mathcal{R}\left(f\left(\mathcal{A}^{-1}(\boldsymbol{z}_k, \boldsymbol{\theta}_G)\right), \boldsymbol{\theta}_I\right) + \underline{m}(\boldsymbol{y}_j), \quad k = 1, \ldots, K \tag{27.27}$$

The task is to estimate the parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_G, \boldsymbol{\theta}_I)$ for the geometric and the radiometric transformation and the unknown true signal f from the observed values $g(\boldsymbol{y}_j)$ and $h(\boldsymbol{z}_k)$.

Symmetric matching referring to the window centres. Often the two windows are approximately aligned, i.e., the centres of the windows approximately match. This is the case if the two windows are chosen around two corresponding keypoints. Then the coordinates $\boldsymbol{y}$ and $\boldsymbol{z}$ refer to the centres of the two windows. Then it also is useful to choose the coordinate system of the unknown signal such that the window is symmetric w.r.t. its centre.

27.3.2 Classical models

27.3.2.1 Stochastical models

The noise model The statistical properties of the noise need to be specified, e.g., assuming the variance to be signal dependent, thus e.g.,

$$\sigma_n^2(\boldsymbol{y}) = T_v(\tilde{g}(\boldsymbol{y}), \boldsymbol{\theta}_V) \qquad \sigma_m^2(\boldsymbol{z}) = T_v(\tilde{h}(\boldsymbol{z}), \boldsymbol{\theta}_V) \tag{27.28}$$

Only for industrial cameras we can expect a linear relation

$$\sigma_n^2(\boldsymbol{y}) = a + b\tilde{g}(\boldsymbol{y}) \qquad \sigma_n^2(\boldsymbol{z}) = a + b\tilde{h}(\boldsymbol{z}) \tag{27.29}$$

resulting from the Poisson distribution of the photon noise.

Prior for the signal f As an example, we might assume the signal to be smooth, i.e., the second derivatives to have expectation 0 and some, possibly inhomogeneous variance. In the simplest case one would have with the local 2×2 Hessian matrix $\mathsf{H} = \mathsf{H}(f(\boldsymbol{x})) = [H_{st}(f(\boldsymbol{x}))]$: $\mathbb{E}(H_{st}) = 0\,, \mathbb{D}(H_{st}(f)) = \sigma_{st}^2\,, s, t, \in \{x, y\}$

27.3.2.2 Radiometric models

A classical model for the intensities is an affine one

$$h = af + b\,. \tag{27.30}$$

A quite general model would assume a monotonic relation

$$h = h(f)\,, \quad \text{with} \quad \frac{\mathrm{d}h}{\mathrm{d}f} > 0\,. \tag{27.31}$$

27.3.2.3 Geometrical models

For describing the classical models for the geometry, we use

$$\boldsymbol{y} = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} x'' \\ y'' \end{bmatrix} \tag{27.32}$$

suggesting the coordinates refer two images.

They are

1. a *pure shift* with $U = 2$ unknown parameters

$$\boldsymbol{y} = \boldsymbol{z} - \boldsymbol{u} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x'' \\ y'' \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \tag{27.33}$$

as the zero-order approximation of all more general geometric transformations.
This model holds rigorously only, in case the two images are taken with identical cameras, show no rotation and the base direction is parallel to the images and the surface is fronto-parallel to the images.

2. an *affinity* with $U = 6$ unknown parameters

$$\boldsymbol{y} = \mathsf{A}\boldsymbol{z} + \boldsymbol{a} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \tag{27.34}$$

as the first order approximation of all more general smooth (differentiable) geometric transformations.
In addition, this model is rigorous only, in case the scene surface is planar in a finite region.

3. an homography with $U = 8$ unknown parameters

$$\mathbf{y} = \mathsf{H}\mathbf{z} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{1}{h_7 x'' + h_8 y'' + h_9} \begin{bmatrix} h_1 x'' + h_2 + y'' + h_3 \\ h_4 x'' + h_5 y'' + h_6 \end{bmatrix} \quad \text{with} \quad |\mathsf{H}| = 1\,. \tag{27.35}$$

This model is rigorous in case the scene is planar within the depicted region.

The next two models result from the assumption that the two images are rectified to epipolar geometry, thus fulfil $y' = y''$.

4. a shift only in the direction of the first coordinate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x'' - u \\ y'' \end{bmatrix} \tag{27.36}$$

This model is rigorous if the scene (the image line) is fronto-parallel to the rectified images.

5. an affine transformation in the direction of the first coordinate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x'' + a_2 y'' \\ y'' \end{bmatrix} \tag{27.37}$$

This model is rigorous in case the scene is locally planar in a small region.
The three parameters a, b and c correspond to the slopes A_1 and A_2 of the plane in X- and Y-direction and the depth A_0.

In case two cameras have projection matrices

$$\mathsf{P}' = \mathsf{K}[I_3 \mid \mathbf{0}]\,, \qquad \mathsf{P}'' = \mathsf{K}[I_3 \mid -\boldsymbol{B}] \tag{27.38}$$

with identical Euclidean cameras and basis in X-direction

$$\mathsf{K} = \begin{bmatrix} c & & \\ & c & \\ & & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} B_X \\ 0 \\ 0 \end{bmatrix} \tag{27.39}$$

and plane

$$Z = A_0 + A_1 X + A_2 Y\,, \qquad \mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ -1 \\ A_0 \end{bmatrix} \tag{27.40}$$

we have

$$a_0 = -\frac{cB_X}{A_0}\,, \qquad a_1 = \frac{A_0 + B_X A_1}{A_0}\,, \qquad a_2 = \frac{B_X A_2}{A_0} \tag{27.41}$$

and inversely

$$A_0 = -\frac{c}{a_0} B_X\,, \qquad A_1 = -\frac{c}{a_0}\,(a_1 - 1)\,, \qquad A_2 = -\frac{c}{a_0} a_2 \tag{27.42}$$

which can be proven using the homography between the two images

$$\mathbf{x}'' = \mathsf{P}''\mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\overline{\mathsf{Q}'}\mathbf{x}' = \begin{bmatrix} A_0 + B_X A_1 & B_X A_2 & -cB_X \\ 0 & A_0 & 0 \\ 0 & 0 & A_0 \end{bmatrix} \mathbf{x}' \tag{27.43}$$

where the image coordinates refer to the principle point.

For testing certain properties or when restricting to one-dimensional signals, e.g., in case we only match epipolar lines, we have the following one-dimensional variants, using the coordinates y and z for the two signals

- Pure shift
$$z = y - u\,. \tag{27.44}$$
- Scale and shift
$$z = a_1 y + a_2\,. \tag{27.45}$$

These models may be applied in conjunction with the linear model for the intensities.

27.4 Examples for models together with their realization

In this section we derive a set of basic models which then are used to evaluate the estimation schemes in detail. In all cases we assume the radiometric model to be an affine transformation with two parameters for contrast and brightness.

These are:

- One-dimensional template matching with a shift only. This will be compared to classical cross-correlation.
- One-dimensional asymmetric image matching with shift and scale.
- One-dimensional symmetric image matching with shift and scale.
- Two-dimensional asymmetric image matching with a shift only.
- Two-dimensional symmetric image matching with an affinity.

27.4.1 One-dimensional template matching

We assume the function $f(x)$ is given analytically. Actually we represent it as a sum of cos-functions, see Sect. 27.6, p.338.

27.4.1.1 The model

At corresponding points we assume the intensity model

$$g = sf + r\,, \tag{27.46}$$

and the geometric model

$$y = x + u\,. \tag{27.47}$$

The observed function then is given by $g(y_i) = sf(x_i) + r + n(y_i)$, or

$$\mathbb{E}\left(\underline{g}(y_i)\right) = sf(y_i - u) + r\,, \qquad \mathbb{D}\left(\underline{g}(y_i)\right) \sim \mathcal{N}(0, \sigma_n^2)\,. \tag{27.48}$$

The range of the observations best is taken symmetrically

$$y_j \in [-J, \ldots, +J]\,. \tag{27.49}$$

Equation (27.48) is a non-linear Gauss–Markov-model with the N observations $\boldsymbol{y}$ and the $U = 3$ unknown parameters $\boldsymbol{\theta}$:

$$N := 2J + 1\,, \quad \boldsymbol{y} := \boldsymbol{g}\,, \quad \boldsymbol{\theta} := \begin{bmatrix} u \\ s \\ r \end{bmatrix}\,, \quad \Sigma_{yy} := \sigma_n^2 I_{2J+1}\,. \tag{27.50}$$

27.4.1.2 The Jacobians

We start from approximate values $\boldsymbol{\theta}^0$. Then the linearized model reads as

$$\Delta y_j := g(y_i) - (s^0 f(y_i - u^0) + r^0) = -s^0 f'(y_i - u^0)\Delta u + f(y_i - u^0)\Delta s + \Delta r\,. \tag{27.51}$$

Hence, the $(2M + 1) \times 3$ Jacobian is

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}} =: A = [\boldsymbol{a}_j^\mathsf{T}] \quad \text{with} \quad \boldsymbol{a}_j^\mathsf{T} = [-s^0 f'(y_i - u^0)\ ,\ f(y_i - u^0)\ ,\ 1]\,. \tag{27.52}$$

27.4.2 One-dimensional asymmetric image matching with shift

27.4.2.1 The functional model

For corresponding points, the geometric model is

$$y = x \quad \text{and} \quad z = y + u\,. \tag{27.53}$$

The radiometric model is

$$g = f \quad \text{and} \quad h = af + b \tag{27.54}$$

We start from the two noisy profiles

$$\begin{aligned} g(y_j) &= f(y_j) + n_g(y_j) \quad \text{with} \quad \mathbb{D}(\underline{g}(y_j)) = \sigma_{g_i}^2 && (27.55)\\ h(z_k) &= af(z_k - u) + b + n_h(z_k) \quad \text{with} \quad \mathbb{D}(\underline{h}(z_k)) = \sigma_{h_k}^2\,. && (27.56) \end{aligned}$$

Since u in general will not be integral, the two sets of observations will not correspond. Following Förstner (1993), given the K observations $h_k, k = 1, \ldots, K$ based on some estimate for u we may generate K corresponding interpolated observations $g(z_k - u)$. We also can use these interpolated observations to obtain estimates for $f(z_k - u)$.

We thus start from the two profiles

$$\begin{aligned} g(y_i) &= f(y_i) + n_g(y_i) && (27.57)\\ h(z_k) &= af(z_k - u) + b + n_h(z_k)\,. && (27.58) \end{aligned}$$

With approximate values u^0, a^0, and b^0 from (27.55) we can derive interpolated values $g(z_k - u_0)$ which fulfill

$$a^0 g(z_k - u^0) + b^0 = a^0 f(z_k - u^0) + b^0 + a^0 n_g(z_k - u^0)\,, \tag{27.59}$$

and linearize (27.56)

$$\begin{aligned} h(z_k) &= h^0(z_k) + \frac{\partial h(z_k)}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} + n_h(y_i) && (27.60)\\ &= \left(a^0 f(z_k - u^0) + b^0\right) + \frac{\partial h(z_k)}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} + n_h(z_k) && (27.61)\\ &= \left(a^0 g(z_k - u^0) + b^0 - a^0 n_g(z_k - u^0)\right) + && (27.62)\\ &\quad \left(-a^0 f'(z_k - u^0)\Delta u + f(z_k - u^0)\Delta a + \Delta b\right) + n_h(z_k)\\ &= \left(a^0 g(z_k - u^0) + b^0\right) + && (27.63)\\ &\quad \left(-a^0 f'(z_k - u^0)\Delta u + f(z_k - u^0)\Delta a + \Delta b\right) + && (27.64)\\ &\quad \left(n_h(y_i) - a^0 n_g(z_k - u^0)\right) \end{aligned}$$

The difference of (27.60) and (27.59) can be simplified using the abbreviations

$$\begin{aligned} \Delta g_k &:= h(z_k) - \left(a^0 g(z_k - u^0) + b^0\right), && (27.65)\\ f'_k &:= f'(z_k - u^0)\,, && (27.66)\\ \Delta n_k &:= n_h(z_k) - a^0 n_g(z_k - u^0) && (27.67) \end{aligned}$$

we obtain the linearized model

$$\boxed{\mathbb{E}\left(\underline{\Delta g}_k\right) = -a^0 \hat{f}'_k \Delta u + \hat{f}_k \Delta a + \Delta b\,, \quad k = 1, \ldots, K\,.} \tag{27.68}$$

by using estimates for the Jacobi-coefficients f_k and f'_k, e.g.,

$$\begin{aligned} \hat{f}_k &= \frac{1}{2}\left(g(z_k - u^0) + \frac{1}{a^0}(h(z_k) - b^0)\right) && (27.69)\\ \hat{f}'_k &= \frac{1}{2}\left(g_x(z_k - u^0) + \frac{1}{a^0} h_x(z_k)\right)\,. && (27.70) \end{aligned}$$

For determining the linearized observations Δg_k interpolation of g is necessary. We implemented cubic and linear interpolation, which often is sufficient.

Remark: Observe, the last two equations may be simplified, which can be done in many ways. On possibility is to only refer to one of the two signals, i.e.,

$$\widehat{f}_k = g(z_k - u^0) \quad \text{and} \quad \widehat{f}'_k = g_x(z_k - u^0) \quad \text{or} \quad \widehat{f}_k = \frac{1}{a^0} h(z_k) \quad \text{and} \quad \widehat{f}'_k = \frac{1}{a^0} h_x(z_k) \tag{27.71}$$

In the second case, often it is recommended to use the central differences

$$\widehat{f}'_k = \frac{1}{a^0} \frac{1}{2\Delta x} \left(h(z_{k+1}) - h(z_{k-1}) \right) . \tag{27.72}$$

$\diamond$

27.4.2.2 The effect of interpolation onto the stochastical model

The Covariance matrix requires some detailed discussion, since the second part $(a^0 g(z_k - u^0) + b^0)$ of the linearized observation refers to an interpolated point of the sequence $g(z_k - u^0)$ as u^0 generally is no integer.

Any interpolation of $g(z_k - u^0)$ has two effects:

1. The variance of the interpolated value changes.
2. Neighbouring interpolated values are correlated.

Both effects depend on the type of interpolation and on the non-integer part of the arguments, see Sect. 27.7. With the remainder $r := x - \lfloor x \rfloor \in [0, 1]$ we obtain

$$\sigma_{\overline{n}(x)} = q \, \sigma_n \, , \tag{27.73}$$

where the factor q depends on the type of interpolation. For linear and cubic interpolation, we have

$$\begin{aligned} q^2_{\text{linear}}(r) &= 1 - 2r + 2r^2 \in \left[\frac{1}{2}, 1 \right] \\ q^2_{\text{cubic}}(r) &= \frac{1}{2} \left[1 - 9\,r^2 + 8\,r^3 + 21\,r^4 - 30\,r^5 + 10\,r^6 \right] \in \left[\frac{9}{64}, \frac{1}{2} \right] \approx [0.14, 0.5] \, . \end{aligned}$$

The correlation between two interpolated signals is less than 50%. For practical reasons we will therefore neglect the correlations, which will lead to suboptimal results, especially to unbiased but less accurate estimation results.

Therefore we use the variances of (27.67)

$$\mathbb{D}\left(\underline{\Delta g}_k \right) = \sigma^2_{h_k} + (a^0)^2 \, q^2(r) \, \sigma^2_{g_k} \, , \tag{27.74}$$

where the variance $\sigma^2_{g_k}$ is taken from the intensity at position $z_k - u^0$.

27.4.3 One-dimensional symmetric image matching with shift and scale

27.4.3.1 The model

We assume the following model.

1. We assume the compound *geometric* mapping is

$$\boldsymbol{z} = \mathsf{B}\boldsymbol{y} \quad \text{or} \quad \begin{bmatrix} z \\ 1 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ 1 \end{bmatrix} \tag{27.75}$$

The compound mapping is partitioned

$$\mathsf{B} = \mathsf{A}^2 \, . \tag{27.76}$$

This reads as (27.21) with

$$\mathsf{A} = \begin{bmatrix} \theta_1 & \theta_2 \\ 0 & 1 \end{bmatrix} \tag{27.77}$$

In Euclidean coordinates we have

$$x = \theta_1 y + \theta_2\,, \qquad z = \theta_1 x + \theta_2 \tag{27.78}$$

$$z = b_1 y + b_2 = \theta_1^2 y + (1+\theta_1)\theta_2 \tag{27.79}$$

as $\theta_1(\theta_1 y + \theta_2) + \theta_2 = \theta_1^2 y + (1+\theta_1)\theta_2$. Thus, we have the relations

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \theta_1^2 \\ (1+\theta_1)\theta_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \sqrt{b_1} \\ b_2/(1+\theta_1) \end{bmatrix} \tag{27.80}$$

2. We assume the observed images g and h to be transformed and noisy version the unknown signal f.
   We again use a symmetric set-up leading to the linear relations valid at corresponding points

$$\tilde{g} \mapsto \tilde{f}: \qquad \tilde{f}(x_i) = \theta_3 \tilde{g}(y_j) + \theta_4 \quad \text{or} \quad \tilde{g}(y_j) = \frac{\tilde{f}(x_i) - \theta_4}{\theta_3} \tag{27.81}$$

$$\tilde{f} \mapsto \tilde{h}: \qquad \tilde{h}(z_k) = \theta_3 \tilde{f}(x_k) + \theta_4 \quad \text{or} \quad \tilde{f}(x_k) = \frac{\tilde{h}(z_k) - \theta_4}{\theta_3} \tag{27.82}$$

   and therefore the mapping from g to h is $\tilde{h} = \theta_3(\theta_3 g + \theta_4) + \theta_4$, hence

$$\tilde{g} \mapsto \tilde{h}: \qquad \tilde{h} = s_1 \tilde{g} + s_2 = \theta_3^2 \tilde{g} + (1+\theta_3)\theta_4\,. \tag{27.83}$$

   Thus, we have the same similar relations between the parameters $\boldsymbol{s}$ and (θ_3, θ_4)

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \theta_3^2 \\ (1+\theta_3)\theta_4 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \sqrt{s_1} \\ s_2/(1+\theta_3) \end{bmatrix}. \tag{27.84}$$

Integrating the geometry and intensity transformation we arrive at the following model, which is generative, i.e., allows to simulate observed images

$$\underline{g}(y_j) = \theta_3^{-1}\left(\tilde{f}(\theta_1 y_j + \theta_2) - \theta_4\right) + \underline{n}(y_j) \tag{27.85}$$

$$\underline{h}(z_k) = \theta_3 \tilde{f}\left(\theta_1^{-1}(z_k - \theta_2))\right) + \theta_4 + \underline{m}(z_k) \tag{27.86}$$

The task is to estimate the parameters $\boldsymbol{\theta}$ for the geometric and the radiometric transformation for describing the signal together with the underlying signal $\tilde{f}$ from the observed values $g(y_j)$ and $h(z_k)$.

27.4.3.2 The Jacobians

We need the Jacobians of g and h w.r.t. the four parameters $\boldsymbol{\theta}$.

We explicitly have

$$\underline{g}(y_j) = \theta_3^{-1}\left(\tilde{f}(\theta_1 y_j + \theta_2) - \theta_4\right) + \underline{n}(y_j) \tag{27.87}$$

$$\approx \frac{1}{\theta_3^a}\left(\tilde{f}(\theta_1^a y_j + \theta_2^a) - \theta_4^a\right) \tag{27.88}$$

$$+\frac{1}{\theta_3^a} f_x(\theta_1^a y_j + \theta_2^a) y_j \Delta\theta_1 \tag{27.89}$$

$$+\frac{1}{\theta_3^a} f_x(\theta_1^a y_j + \theta_2^a) \Delta\theta_2 \tag{27.90}$$

$$-\frac{1}{\theta_3^{a,2}}(f(\theta_1 y_j + \theta_2) - \theta_4)\Delta\theta_3 \tag{27.91}$$

$$-\frac{1}{\theta_3^a}\Delta\theta_4 \tag{27.92}$$

$$+\underline{n}(y_j) \tag{27.93}$$

or with

$$f_j := f(\theta_1 y_j + \theta_2) \quad \text{and} \quad f_{xj} := f_x(\theta_1^a y_j + \theta_2^a) \tag{27.94}$$

the corresponding Jacobian

$$\frac{\partial g_j}{\partial \boldsymbol{\theta}} = \left[\frac{1}{\theta_3^a} f_{xj} y_j \mid \frac{1}{\theta_3^a} f_{xj} \mid -\frac{1}{\theta_3^{a,2}} (f_j - \theta_4) \mid -\frac{1}{\theta_3^a} \right] \tag{27.95}$$

or

$$\boxed{\frac{\partial g_j}{\partial \boldsymbol{\theta}} = \frac{1}{\theta_3^{a,2}} \left[\theta_3^a f_{xj} y_j \mid \theta_3^a f_{xj} \mid -(f_j - \theta_4) \mid -\theta_3^a \right]} \tag{27.96}$$

Similarly we have

$$\begin{aligned}
\underline{h}(z_k) &= \theta_3 \tilde{f}\left(\theta_1^{-1}(z_k - \theta_2))\right) + \theta_4 + \underline{m}(z_k) && (27.97)\\
&\approx \theta_3^a \tilde{f}\left(\frac{1}{\theta_1^a}(z_k - \theta_2^a))\right) + \theta_4^a && (27.98)\\
&\quad -\theta_3^a f_x\left(\frac{1}{\theta_1^a}(z_k - \theta_2^a)\right) \frac{z_k - \theta_2}{\theta_1^{a,2}} \Delta\theta_1 && (27.99)\\
&\quad -\theta_3^a f_x\left(\frac{1}{\theta_1^a}(z_k - \theta_2^a))\right) \frac{1}{\theta_1^a} \Delta\theta_2 && (27.100)\\
&\quad +\tilde{f}\left(\theta_1^{-1}(z_k - \theta_2))\right) \Delta\theta_3 && (27.101)\\
&\quad +\Delta\theta_4 && (27.102)\\
&\quad +\underline{m}(z_k) && (27.103)
\end{aligned}$$

With

$$f_k := f\left(\theta_1^{-1}(z_k - \theta_2))\right) \quad \text{and} \quad f_{xk} := f_x\left(\theta_1^{-1}(z_k - \theta_2))\right) \tag{27.104}$$

the corresponding Jacobian are

$$\frac{\partial h_k}{\partial \boldsymbol{\theta}} = \left[-\theta_3^a f_{xk} \frac{z_k - \theta_2^a}{\theta_1^{a,2}} \mid -\theta_3^a f_{xk} \frac{1}{\theta_1^a} \mid f_k \mid 1 \right] \tag{27.105}$$

In order to obtain analogy to the 2D solution we use the substitutions

$$x'_k = \frac{(z_k - \theta_2^a)}{\theta_1^a} \quad \text{and} \quad \phi_{xk} = \frac{f_{xk}}{\theta_1^a} \tag{27.106}$$

and obtain

$$\boxed{\frac{\partial h_k}{\partial \boldsymbol{\theta}} = \left[-\theta_3^a \phi_{xk} x'_k \mid -\theta_3^a \phi_{xk} \mid f_k \mid 1 \right]} \tag{27.107}$$

27.4.3.3 Estimating the signal f

The signal f can be estimated as weighted sum of

$$f^g(x_i) = \widehat{\theta}_3 g\left(\frac{x_i - \widehat{\theta}_2}{\widehat{\theta}_1}\right) + \widehat{\theta}_4 \quad \text{and} \quad f^h(x_i) = \frac{h(\widehat{\theta}_1 x_i + \widehat{\theta}_2) - \widehat{\theta}_4}{\widehat{\theta}_3} \tag{27.108}$$

This requires interpolation of the original observations, which needs to be taken into account. The variances therefore are

$$\mathbb{D}(\underline{f}^g(x_i)) = \widehat{\theta}_3^2 \, q^2(r^g) \, \sigma_{g_i}^2 \quad \text{and} \quad \mathbb{D}(\underline{f}^h(x_i)) = \frac{1}{\widehat{\theta}_3^2} \, q^2(r^h) \, \sigma_{h_i}^2 \,, \tag{27.109}$$

where the variances ($\sigma_{g_i}^2$ and $\sigma_{h_i}^2$) and the remainders,

$$r^g = \frac{x_i - \widehat{\theta}_2}{\widehat{\theta}_1} - \left\lfloor \frac{x_i - \widehat{\theta}_2}{\widehat{\theta}_1} \right\rfloor \quad \text{and} \quad r^h = \widehat{\theta}_1 x_i + \widehat{\theta}_2 - \left\lfloor \widehat{\theta}_1 x_i + \widehat{\theta}_2 \right\rfloor \,, \tag{27.110}$$

are to be taken at the corresponding position. Using the inverse variances as weights, leads to unbiased estimates for $f(x_i)$.

If the noise variances in g and h are the same, and the difference of the factors q is neglected we arrive at the estimated signal

$$\hat{f}(x_i) = \frac{f^g(x_i) + \hat{\theta}_3^4 f^h(x_i)}{1 + \hat{\theta}_3^4}\,. \tag{27.111}$$

27.4.3.4 On the covariance matrix of the estimates

The theoretical covariance matrix derived from the symmetric matching procedure assumes the signal is known, thus in general will be too optimistic. Förstner (1998) provides a rigorous analysis of multi-image matching with unknown shift, however, with identical weights for all images. The normal equation system contains three blocks, (1) for the K sets of shifts $[\boldsymbol{p}_1; ...; \boldsymbol{p}_K]$, (2) for the unknown signal, and (3) for the Lagrangian multipliers which fix the gauge, namely requiring the sum of all shifts is zero.

For two images with $W_1 = w_1 I$ and $W_2 = w_2 I$ the general normal equation matrix specializes to

$$\begin{bmatrix} w_1 N_0 & 0 & w_1 A^\mathsf{T} & w_1 I_2 \\ 0 & w_2 N_0 & w_2 A^\mathsf{T} & w_2 I_2 \\ w_1 A & w_2 A & w_1 + w_2 & 0 \\ w_1 I_2 & w_2 I_2 & 0^\mathsf{T} & 0 \end{bmatrix} \tag{27.112}$$

where

$$N_0 = \sum_i (\nabla f_i \nabla^\mathsf{T} f_i)\,. \tag{27.113}$$

The inverse is

$$\frac{1}{w_1 + w_2} \begin{bmatrix} w_2/w_1 N_0^{-1} & -N_0^{-1} & 0 & I_2 \\ -N_0^{-1} & w_1/w_2 N_0^{-1} & 0 & I_2 \\ 0 & 0 & I_I & -A \\ I_2 & I_2 & -A^\mathsf{T} & 0 \end{bmatrix} \tag{27.114}$$

Which can be checked by $N N^{-1} = I$, or

$$= \frac{1}{w_1 + w_2} \begin{bmatrix} w_2 I + w_1 I & -w_2 I + w_2 I & 0 + 0 & w_2 N_0^{-1} - w_2 N_0^{-1} \\ - & w_1 I + w_2 I & -w_1 A^\mathsf{T} N_0^{-1} + w_1 A^\mathsf{T} N_0^{-1} & -w_2 N_0^{-1} + w_2 N_0^{-1} \\ - & - & (w_1 + w_2) I_I & 0 \\ - & - & - & (w_1 + w_2) I_2 \end{bmatrix} \tag{27.115}$$

Therefore, the precision of the shift is

$$\Sigma_{\widehat{\Delta p}\widehat{\Delta p}} := \boldsymbol{D}(\hat{\boldsymbol{p}}_2 - \hat{\boldsymbol{p}}_1) = \frac{w_1 + w_2}{w_1 w_2} N_0^{-1} \tag{27.116}$$

If the signal is not estimated simultaneously but taken to be fixed and we solve the normal equation system in a symmetric manner

$$(w_1 + w_2) N_0 \, \frac{\widehat{\Delta \boldsymbol{p}}}{2} = \boldsymbol{h} \tag{27.117}$$

and obtain the covariance matrix for the difference $\widehat{\Delta \boldsymbol{p}}$

$$\Sigma^*_{\widehat{\Delta p}\widehat{\Delta p}} = \frac{4}{w_1 + w_2} N_0^{-1} \tag{27.118}$$

Hence, the true covariance matrix $\Sigma_{\widehat{\Delta p}\widehat{\Delta p}}$ is a factor

$$\boxed{k = \frac{\mathrm{tr}\Sigma_{\widehat{\Delta p}\widehat{\Delta p}}}{\mathrm{tr}\Sigma^*_{\widehat{\Delta p}\widehat{\Delta p}}} = \frac{(w_1 + w_2)^2}{4 w_1 w_2}} \tag{27.119}$$

larger than the too optimistic covariance matrix $\Sigma^*_{\widehat{\Delta p}\widehat{\Delta p}}$ from the simplified model. However, if the two weights are equal $w_1 = w_2$, the covariance matrix derived from the approximate model is identical to the covariance matrix from the rigorous model.

It is an open question, whether this factor also holds if more than the shift parameters are estimated. It certainly is difficult to generalize it to the case where the individual pixels have different noise standard deviations.

We will use this factor by replacing the weights w_1 and w_2 by the average weight in the two images.

27.4.4 Two-dimensional template matching

We assume the geometric model

$$\mathbf{z} = \mathsf{A}\mathbf{x} \quad \text{or} \quad \boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{a} \tag{27.120}$$

with the parametrization

$$\mathsf{A} = \begin{bmatrix} A & \boldsymbol{a} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_3 & \theta_5 \\ \theta_2 & \theta_4 & \theta_6 \\ 0 & 0 & 1 \end{bmatrix} . \tag{27.121}$$

Similarly, we have the affine intensity model for the observed signal h

$$h = \theta_7 f + \theta_8 . \tag{27.122}$$

– in order to be consistent with the symmetric image matching model later. This model thus has 8 unknown parameters. Thus we explicitly have

$$h_k = \theta_7 \, f(\boldsymbol{x}_k) + \theta_8 \quad \text{with} \quad \boldsymbol{x}_k = A^{-1}\boldsymbol{z}_k + \boldsymbol{a} . \tag{27.123}$$

27.4.5 Two-dimensional asymmetric image matching with shift only

Similarly to Sect. 27.4.2, we start from the two noisy profiles

$$g(\boldsymbol{x}_j) = f(\boldsymbol{x}_j) + n_g(\boldsymbol{y}_j) \tag{27.124}$$
$$h(\boldsymbol{z}_k) = f(\boldsymbol{z}_k - \boldsymbol{u}) + n_h(\boldsymbol{z}_k) \tag{27.125}$$

where

$$\boldsymbol{x}_i = \begin{bmatrix} x \\ y \end{bmatrix}_i \quad \text{and} \quad \boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix} . \tag{27.126}$$

Using approximate values $\boldsymbol{u}^0$ and the abbreviations

$$\Delta g_k := h(\boldsymbol{z}_k) - g(\boldsymbol{z}_k - \boldsymbol{u}^0) , \tag{27.127}$$
$$f'_k := f'(\boldsymbol{z}_k - \boldsymbol{u}^0) , \tag{27.128}$$
$$\overline{n}_k := n_h(\boldsymbol{z}_k) - n_g(\boldsymbol{z}_k - \boldsymbol{u}^0) \tag{27.129}$$

we arrive at the linearized model

$$\Delta g_k = -\widehat{f}_{xk}\widehat{\Delta u} - \widehat{f}_{yk}\widehat{\Delta v} + \overline{n}_k \tag{27.130}$$

with the estimates

$$\widehat{f}_k = \frac{1}{2}\left(g(\boldsymbol{z}_k - \boldsymbol{u}^0) + h(\boldsymbol{z}_k)\right) \tag{27.131}$$
$$\widehat{f}_{xk} = \frac{1}{2}\left(g_x(\boldsymbol{z}_k - \boldsymbol{u}^0) + h_x(\boldsymbol{z}_k)\right) \tag{27.132}$$
$$\widehat{f}_{yk} = \frac{1}{2}\left(g_y(\boldsymbol{z}_k - \boldsymbol{u}^0) + h_y(\boldsymbol{z}_k)\right) . \tag{27.133}$$

Obviously, we need to interpolate g and its partial derivatives if $\boldsymbol{u}^0$ is not integer.

The variance of the observations is

$$\sigma^2_{\overline{n}_k} = \mathbb{D}\left(\underline{n}_g(\boldsymbol{z}_k - \boldsymbol{u}^0)\right) + \mathbb{D}\left(\underline{n}_h(\boldsymbol{z}_k)\right) . \tag{27.134}$$

Again, we need to perform variance propagation for the interpolated values. We the reminder for arbitrary $\boldsymbol{x} := \boldsymbol{z}_k - \boldsymbol{u}^0$

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} x - \lfloor x \rfloor \\ y - \lfloor y \rfloor \end{bmatrix} \tag{27.135}$$

and have

$$\begin{aligned} g(x,y) \quad = \quad & (1-r)(1-s)\, g(\lfloor x \rfloor, \lfloor y \rfloor) + && (27.136)\\ & (1-r)s\, g(\lfloor x \rfloor, \lfloor y \rfloor + 1) + && (27.137)\\ & (r(1-s)\, g(\lfloor x \rfloor + 1, \lfloor y \rfloor) + && (27.138)\\ & rs\, g(\lfloor x \rfloor + 1, \lfloor y \rfloor + 1) . && (27.139) \end{aligned}$$

Assuming the intensities have the same variance σ_g^2 we obtain the variance of the interpolated value

$$\mathbb{D}\left(n_g(\boldsymbol{z}_k - \boldsymbol{u}^0)\right) = ((1-r)^2 + r^2)((1-s)^2 + s^2)\sigma_g^2 . \tag{27.140}$$

On an average we obtain a variance of

$$\overline{\mathbb{D}\left(n_g\right)} = \frac{4}{9}\sigma_g^2 . \tag{27.141}$$

Hence, we either can use the individual value using (27.140) hence

$$\sigma^2_{\overline{n}_k} = ((1-r)^2 + r^2)((1-s)^2 + s^2)\mathbb{D}(\underline{n}_g(\boldsymbol{z}_k - \boldsymbol{u}^0)) + \mathbb{D}(\underline{n}_h(\boldsymbol{z}_k)) \tag{27.142}$$

or the average value

$$\overline{\sigma^2_{\overline{n}_k}} = \int_{r=0}^{1}\int_{s=0}^{1} \sigma^2_{\overline{n}_k} \mathrm{d}r\mathrm{d}s + \mathbb{D}(\underline{n}_h(\boldsymbol{z}_k)) = \frac{4}{9}\mathbb{D}(\underline{n}_g(\boldsymbol{z}_k - \boldsymbol{u}^0)) + \mathbb{D}(\underline{n}_h(\boldsymbol{z}_k)) \tag{27.143}$$

in both cases assuming the neighbouring pixels have the same noise variance. If the noise is homogeneous in both images, the noise variance $\sigma^2_{\overline{n}}$ also is constant, since the reminders (r, s) are the same for all pixels.

The normal equations read as

$$\boldsymbol{N}\widehat{\Delta\boldsymbol{\theta}} = \boldsymbol{n} \tag{27.144}$$

with the weights

$$w_k = \sigma^{-2}_{\overline{n}_k} , \tag{27.145}$$

the unknown vector

$$\widehat{\Delta\boldsymbol{\theta}} = \begin{bmatrix} \widehat{\Delta u} \\ \widehat{\Delta v} \end{bmatrix} \tag{27.146}$$

the normal equation matrix

$$\boldsymbol{N} = \begin{bmatrix} \sum_k w_k \hat{f}^2_{xk} & \sum_k w_k \hat{f}_{xk}\hat{f}_{yk} \\ \sum_k w_k \hat{f}_{xk}\hat{f}_{yk} & \sum_k w_k \hat{f}^2_{yk} \end{bmatrix} \tag{27.147}$$

and the right hand side

$$\boldsymbol{n} = -\begin{bmatrix} \sum_k w_k \hat{f}_{xk}\Delta g_k \\ \sum_k w_k \hat{f}_{yk}\Delta g_k \end{bmatrix} . \tag{27.148}$$

If the noise variance in both images is assumed to be homogeneous, the weights can be replaced by a constant value $w = 1/\sigma^2_{\overline{n}}$. Furthermore, for the final evaluation it is of advantage to use the variance $\sigma^2_{\overline{n}}$ from (27.142), since it is the same for all pixels.

27.4.6 Two-dimensional symmetric image matching with affinity

We now address the symmetric matching of two two-dimensional signals geometrically related by an affine transformation. Hence, we have the geometric model

$$\mathbf{y} \mapsto \mathbf{x}: \quad \mathbf{x} = \mathsf{A}\mathbf{y} \tag{27.149}$$
$$\mathbf{x} \mapsto \mathbf{z}: \quad \mathbf{z} = \mathsf{A}\mathbf{x} \tag{27.150}$$

or

$$\boldsymbol{y} \mapsto \boldsymbol{x}: \quad \boldsymbol{x} = \mathsf{A}\boldsymbol{y} + \boldsymbol{a} \tag{27.151}$$
$$\boldsymbol{x} \mapsto \boldsymbol{z}: \quad \boldsymbol{z} = \mathsf{A}\boldsymbol{x} + \boldsymbol{a} \tag{27.152}$$

We use the parametrization

$$\mathsf{A} = \begin{bmatrix} \mathsf{A} & \boldsymbol{a} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_3 & \theta_5 \\ \theta_2 & \theta_4 & \theta_6 \\ 0 & 0 & 1 \end{bmatrix} . \tag{27.153}$$

Similarly, we have the affine intensity model

$$g \mapsto f: \quad f = \theta_7 g + \theta_8 \tag{27.154}$$
$$f \mapsto h: \quad h = \theta_7 f + \theta_8 , \tag{27.155}$$

as in (27.81). This model thus has 8 unknown parameters.

27.4.6.1 Jacobians of g

We start with the differential for

$$g_j = \theta_7^{-1} \left(f(\boldsymbol{x}_j) - \theta_8 \right) \quad \text{with} \quad \boldsymbol{x}_j = \mathsf{A}\boldsymbol{y}_j + \boldsymbol{a} \tag{27.156}$$

w.r.t. to the unknown parameters $\boldsymbol{\theta}$. We explicitly have

$$g_j = \theta_7^{-1} \left(f(\theta_1 x_j + \theta_3 y_j + \theta_5, \theta_2 x_j + \theta_4 y_j + \theta_6) - \theta_8 \right) . \tag{27.157}$$

Due to $\mathrm{d}(1/x) = -1/x^2 \mathrm{d}x$ we explicitly have

$$g_j = \left\{ \theta_7^{-1} \left(\underbrace{f(\mathsf{A}\boldsymbol{y}_j + \boldsymbol{a})}_{f_j} - \theta_8 \right) \right\}^a \tag{27.158}$$
$$+\theta_7^{-1} f_{x,j} x_j \, \mathrm{d}\theta_1 \tag{27.159}$$
$$+\theta_7^{-1} f_{y,j} x_j \, \mathrm{d}\theta_2 \tag{27.160}$$
$$+\theta_7^{-1} f_{x,j} y_j \, \mathrm{d}\theta_3 \tag{27.161}$$
$$+\theta_7^{-1} f_{y,j} y_j \, \mathrm{d}\theta_4 \tag{27.162}$$
$$+\theta_7^{-1} f_{x,j} \, \mathrm{d}\theta_5 \tag{27.163}$$
$$+\theta_7^{-1} f_{y,j} \, \mathrm{d}\theta_6 \tag{27.164}$$
$$-\theta_7^{-2} (f_j - \theta_8) \, \mathrm{d}\theta_7 \tag{27.165}$$
$$-\theta_7^{-1} \, \mathrm{d}\theta_8 \tag{27.166}$$

27.4.6.2 Jacobians of h

For the derivatives of

$$h_k = \theta_7 \underbrace{f(\boldsymbol{x}_k)}_{f_k} + \theta_8 \quad \text{with} \quad \boldsymbol{x}_k = \mathsf{A}^{-1}(\boldsymbol{z}_k - \boldsymbol{a}) \tag{27.167}$$

w.r.t. $\boldsymbol{\theta}$ we need some preparation. The Jacobian of the inverse is

$$\mathrm{d}A^{-1} = -A^{-1}(\mathrm{d}A)A^{-1} \quad \text{with} \quad A^{-1} = \frac{1}{D}\begin{bmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{bmatrix} \quad \text{and} \quad D := |A|\,. \tag{27.168}$$

We in addition use the substitution

$$\boldsymbol{x}_k = A^{-1}(\boldsymbol{z}_k - \boldsymbol{a})\,. \tag{27.169}$$

These are the coordinates of the point in f corresponding to $\boldsymbol{z}_k$. Hence, with $[\alpha_{ij}] = A^{-1}$ we have

$$\begin{aligned}
\mathrm{d}(A^{-1}(\boldsymbol{z}_k - \boldsymbol{a})) &= -A^{-1}(\mathrm{d}A)A^{-1}(\boldsymbol{z}_k - \boldsymbol{a}) && (27.170)\\
&= -A^{-1}(\mathrm{d}A)\boldsymbol{z}'_k && (27.171)\\
&= -(\boldsymbol{z}'^{\mathsf{T}}_k \otimes A^{-1})\mathrm{d}(\mathrm{vec}A) && (27.172)\\
&= -\begin{bmatrix} \alpha_{11}x'_k & \alpha_{12}x'_k & \alpha_{11}y'_k & \alpha_{12}y'_k \\ \alpha_{21}x'_k & \alpha_{22}x'_k & \alpha_{21}x'_k & \alpha_{22}x'_k \end{bmatrix}. && (27.173)
\end{aligned}$$

We finally have

$$\begin{aligned}
\frac{\partial f(\boldsymbol{z}'_k)}{\partial \boldsymbol{\theta}} &= \frac{\partial f(\boldsymbol{z}'_k)}{\partial x'_k}\frac{\partial x'_k}{\partial \boldsymbol{\theta}} + \frac{\partial f(\boldsymbol{z}'_k)}{\partial y'_k}\frac{\partial y'_k}{\partial \boldsymbol{\theta}} && (27.174)\\
&= - \quad \nabla^{\mathsf{T}} f\ (\boldsymbol{z}'^{\mathsf{T}}_k \otimes A^{-1}) && (27.175)\\
&= -(1 \otimes \nabla^{\mathsf{T}} f)(\boldsymbol{z}'^{\mathsf{T}}_k \otimes A^{-1}) && (27.176)\\
&= -\boldsymbol{z}'^{\mathsf{T}}_k \otimes \nabla^{\mathsf{T}} f A^{-1} && (27.177)
\end{aligned}$$

Therefore with

$$\nabla\phi = \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} = A^{-\mathsf{T}}\nabla f \tag{27.178}$$

we obtain

$$\frac{\partial f(\boldsymbol{z}'_k)}{\partial \boldsymbol{\theta}} = -\boldsymbol{z}'^{\mathsf{T}}_k \otimes \nabla^{\mathsf{T}}\phi_k = -[x'_k\phi_{x,k} \quad x'_k\phi_{y,k} \quad y'_k\phi_{x,k} \quad y'_k\phi_{y,k}]\,. \tag{27.179}$$

Finally, the differential of h is

$$\begin{aligned}
h_k &= \left\{ \theta_7 \underbrace{f(A^{-1}(\boldsymbol{z}_k - \boldsymbol{a}))}_{f_k} + \theta_8 \right\}^a && (27.180)\\
&\quad -\theta_7\phi_{x,k}x'_k\,\mathrm{d}\theta_1 && (27.181)\\
&\quad -\theta_7\phi_{y,k}x'_k\,\mathrm{d}\theta_2 && (27.182)\\
&\quad -\theta_7\phi_{x,k}y'_k\,\mathrm{d}\theta_3 && (27.183)\\
&\quad -\theta_7\phi_{y,k}y'_k\,\mathrm{d}\theta_4 && (27.184)\\
&\quad -\theta_7\phi_{x,k}\,\mathrm{d}\theta_5 && (27.185)\\
&\quad -\theta_7\phi_{y,k}\,\mathrm{d}\theta_6 && (27.186)\\
&\quad +f_k\,\mathrm{d}\theta_7 && (27.187)\\
&\quad +1\,\mathrm{d}\theta_8 && (27.188)
\end{aligned}$$

27.5 Generating approximate values with a prespecified distance to the true values

When generating artificial data, we need to generate parameter values with certain properties:

- The symmetry of the algorithms likely depends on the deviation of the parameters from the unit transformation.
- The convergence of the algorithms depends on the deviations of the approximate values from the true values of the parameters.

In both cases we want to generate parameters $\boldsymbol{\theta}$ which differ from some reference parameters $\boldsymbol{\theta}_0$ by a specified amount. Since, the units of the parameters are different we measure the difference between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$ by the Mahalanobis distance:

$$d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = ||\boldsymbol{\theta} - \boldsymbol{\theta}_0||_\Sigma = \sqrt{(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\mathsf{T} \Sigma_{\theta\theta}^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)}\,. \tag{27.189}$$

Using this distance, we now are able to generate random deviations $\underline{\underline{\Delta\boldsymbol{\theta}}}$ with a prespecified distance d_s, which now is measured in units of standard deviations and taking the correlations into account. With a random 8-vector

$$\underline{\boldsymbol{e}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\theta\theta})\,, \tag{27.190}$$

we obtain for a specific sample $\boldsymbol{e}$:

$$\Delta\boldsymbol{\theta} = \frac{d_s}{d(\boldsymbol{e}, \mathbf{0})}\, \boldsymbol{e}\,, \tag{27.191}$$

such that

$$d(\Delta\boldsymbol{\theta}, \mathbf{0}) = d_s\,. \tag{27.192}$$

Starting from some reference parameter vector $\boldsymbol{\theta}_0$ we, in order to ensure that the resulting vector $\boldsymbol{\theta}$ represents a valid transformation we apply the change $\Delta\boldsymbol{\theta}$ multiplicatively:

$$A(\boldsymbol{\theta}) = \exp(K_A(\Delta\boldsymbol{\theta}))\; A(\boldsymbol{\theta}_0) \quad \text{with} \quad K_A(\Delta\boldsymbol{\theta}) = \begin{bmatrix} \Delta\theta_1 & \Delta\theta_3 & \Delta\theta_5 \\ \Delta\theta_2 & \Delta\theta_4 & \Delta\theta_6 \\ 0 & 0 & 0 \end{bmatrix} \tag{27.193}$$

and similarly

$$R(\boldsymbol{\theta}) = \exp(K_R(\Delta\boldsymbol{\theta}))\; R(\boldsymbol{\theta}_0) \quad \text{with} \quad K_R(\Delta\boldsymbol{\theta}) = \begin{bmatrix} \Delta\theta_7 & \Delta\theta_8 \\ 0 & 0 \end{bmatrix}. \tag{27.194}$$

The covariance matrix $\Sigma_{\theta\theta}$ can be taken from the generated true image.

Sometimes we might to change the geometric and the radiometric affinity separately, specifying individual distances d_{sA} and d_{sR}. Then we replace (27.191) by

$$\Delta\boldsymbol{\theta} = \begin{bmatrix} \dfrac{d_{sA}}{d(\boldsymbol{e}_A, \mathbf{0})}\, \boldsymbol{e}_A \\ \dfrac{d_{sR}}{d(\boldsymbol{e}_R, \mathbf{0})}\, \boldsymbol{e}_R \end{bmatrix}, \tag{27.195}$$

where the indices A and R refer to the first 6 and the last 2 elements of the 8-vectors.

27.6 Generating true functions having a certain roughness

27.6.1 Generating 1D functions

For testing we need a smooth function f. In order to be able to vary its roughness we assume its power spectrum is a central Gaussian with a prespecified standard deviation b.

Hence, we represent the smooth function as

$$f(x) = \int_{u=-\infty}^{+\infty} \breve{f}(u; b)\; \mathrm{e}^{+\mathrm{i}2\pi u x} \mathrm{d}u\,. \tag{27.196}$$

and assume the amplitude spectrum $\breve{f} := \mathcal{F}(f)$ with random phase

$$\breve{f}(u;b) = G(0,b^2)\mathrm{e}^{\mathrm{i}2\pi e_u} = \frac{1}{\sqrt{2\pi b^2}}\,\mathrm{e}^{-\frac{1}{2}\left(\frac{u}{b}\right)^2}\,\mathrm{e}^{\mathrm{i}2\pi e_u}\,. \tag{27.197}$$

with

$$\underline{e}_u \sim \mathcal{U}(0,1)\,. \tag{27.198}$$

The standard deviation b can be called the (effective) bandwidth, see McGillem and Svedlow (1976): The signal on an average will show wavelengths $1/b$.

We replace the integral by a finite sum taken over a random K-sample $\breve{f}_k(u_k)$ with $(u_1, u_2, ..., u_K)$ of frequencies u_k with $u_1 = 0$ and

$$u_k \sim \mathcal{N}(0,b^2)\,, \quad k = 2,\ldots,K\,. \tag{27.199}$$

In order to achieve a prespecified mean μ_f and a prespecified standard deviation σ_f, and obtain a real valued function f we

- choose the value $\breve{f}(0) := \breve{f}_1(0) = \mu_f$
- choose the norm of the other values is σ_f:

$$\sum_{k=2}^{K} ||\breve{f}_k||^2 = \sigma_f^2\,; \tag{27.200}$$

- eliminate the imaginary parts by taking the mean of the signal and its conjugate.

Therefore we determine the real valued signal with standard deviation b using the random frequencies from (27.199)

$$f(x) = \frac{1}{2}\sum_{k=1}^{K}\left(\breve{f}_k\,\mathrm{e}^{+\mathrm{i}2\pi u_k x} + \breve{f}_k^*\,\mathrm{e}^{-\mathrm{i}2\pi u_k x}\right)\,. \tag{27.201}$$

For an N-vector of positions $\boldsymbol{x}$ we realize it by determining the transposed vector of the matrix F,

$$\boldsymbol{f}^\mathsf{T} = f(\boldsymbol{x}) = \frac{1}{2}\mathbf{1}_K^\mathsf{T}\left((\breve{\boldsymbol{f}}\,\mathbf{1}_N^\mathsf{T}) \odot \mathrm{e}^{+\mathrm{i}2\pi\boldsymbol{u}\boldsymbol{x}^\mathsf{T}} + (\breve{\boldsymbol{f}}^*\,\mathbf{1}_N^\mathsf{T}) \odot \mathrm{e}^{-\mathrm{i}2\pi\boldsymbol{u}\boldsymbol{x}^\mathsf{T}}\right) \tag{27.202}$$

where $\mathrm{e}^M = [\mathrm{e}^{m_{ij}}]$ is the elementwise exponential.

27.6.2 Generating 2D functions

Here we represent the smooth function as

$$f(\boldsymbol{x}) = \int_{\boldsymbol{u}=-\infty}^{+\infty} \breve{f}(\boldsymbol{u};b)\,\mathrm{e}^{+\mathrm{i}2\pi\boldsymbol{u}^\mathsf{T}\boldsymbol{x}}\mathrm{d}\boldsymbol{u}\,. \tag{27.203}$$

and assume the amplitude spectrum with random phase

$$\breve{f}(\boldsymbol{u};b) = G(\mathbf{0};b^2)\mathrm{e}^{\mathrm{i}2\pi e_u} = \frac{1}{\sqrt{2\pi b^2}}\,\mathrm{e}^{-\frac{1}{2}\frac{||\boldsymbol{u}||^2}{b^2}}\,\mathrm{e}^{\mathrm{i}2\pi e_u}\,. \tag{27.204}$$

with

$$\underline{e}_u \sim \mathcal{U}(0,1) \quad \text{except for} \quad \boldsymbol{u} = \mathbf{0}\,. \tag{27.205}$$

Again we generate the K-vector $\breve{\boldsymbol{f}} = [\breve{f}(\boldsymbol{u}_k)]$ by choosing $K-1$ random values $\boldsymbol{u}_k \sim \mathcal{N}(\mathbf{0},b^2), k = 2, ..., K$, and normalize $\breve{\boldsymbol{f}}$ such that $\sum_k ||\breve{f}_k||^2 = \sigma_f^2$ and choose $\breve{f}(\boldsymbol{u}_1) = \mu_f$ for $\boldsymbol{u}_1 = [0,0]$. The we obtain the signal from

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{k=1}^{K}\left(\breve{f}_k\,\mathrm{e}^{+\mathrm{i}2\pi\boldsymbol{u}_k^\mathsf{T}\boldsymbol{x}} + \breve{f}_k^*\,\mathrm{e}^{-\mathrm{i}2\pi\boldsymbol{u}_k^\mathsf{T}\boldsymbol{x}}\right)\,. \tag{27.206}$$

For two $N \times N$-matrices X and Y of positions with N^2-vectors $\boldsymbol{x} = \mathrm{vec}\mathsf{X}$ and $\boldsymbol{y} = \mathrm{vec}\mathsf{Y}$, the $K \times 2$-matrix $\mathsf{U} = [\boldsymbol{u}_x, \boldsymbol{u}_y]$ we realize it by

$$\mathrm{vec}(\mathsf{F})^\mathsf{T} = f(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\mathbf{1}_K^\mathsf{T} \Big((\check{\boldsymbol{f}}\, \mathbf{1}_{N^2}^\mathsf{T}) \odot \mathrm{e}^{+\mathrm{i}2\pi(\boldsymbol{u}_x\boldsymbol{x}^\mathsf{T} + \boldsymbol{u}_y\boldsymbol{y}^\mathsf{T})} \tag{27.207}$$

$$+ (\check{\boldsymbol{f}}^*\, \mathbf{1}_{N^2}^\mathsf{T}) \odot \mathrm{e}^{-\mathrm{i}2\pi(\boldsymbol{u}_x\boldsymbol{x}^\mathsf{T} + \boldsymbol{u}_y\boldsymbol{y}^\mathsf{T})} \Big) \tag{27.208}$$

27.7 Smoothing and Interpolating

The function $f(x, y)$ needs to be C_1-smooth, i.e., the function values and the first derivatives should be continuous. Otherwise, the iteration process might lead to jumps, which themselves might lead to oscillations.

This can be achieved in various ways. We discuss

- Bilinear interpolation, due to its simplicity, which, however, is not C_1-smooth.
- Quadratic interpolation
- Cubic interpolation
- B-splines

We also discuss the 2D case.

In all cases we provide a compact representation and derive the effect on noisy data, especially the reduction of the noise variance caused by smoothing and interpolation. This is straight forward, since all methods are linear operators.

27.7.1 Linear interpolation

27.7.1.1 1D linear interpolation

Compact Representation. Let the signal be given by a sequence b_m. Let us assume the function values are $f(m) = b_m$. Hence, we assume the pixel distance is $\Delta x = 1$. Linear interpolation at position x refers to the reference point

$$i = \lfloor x \rfloor\,, \tag{27.209}$$

and its neighbour $i + 1$. It is a linear function of

$$u = u(x) = x - i\,. \tag{27.210}$$

We explicitly have

$$f(x) = (1 - u)f(i) + uf(i + 1) = f(i) + uf(i) + uf(i + 1)\,. \tag{27.211}$$

This can be written as

$$f(x) = [1 \mid u] \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} f(i) \\ f(i+1) \end{bmatrix}\,. \tag{27.212}$$

or with the substitutions

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ u \end{bmatrix}\,, \qquad \mathsf{M}_0 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}\,, \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} b_i \\ b_{i+1} \end{bmatrix} \tag{27.213}$$

as

$$\boxed{f(x) = \boldsymbol{u}^\mathsf{T}(x)\mathsf{M}_0\boldsymbol{b}\,.} \tag{27.214}$$

This compact notation can be transferred to higher order polynomials and also to the determination of derivatives, where we will use matrices M_d, where d indicates the derivative.

Variance and covariance of linear interpolation. We first derive variances of linearly interpolated values.

Linear interpolation at $x \in [x_i, x_i + 1]$ uses the partitioning

$$x = x_i + r \quad \text{with} \quad x_i = \lfloor x \rfloor \quad \text{and} \quad r = x - x_i \,. \tag{27.215}$$

We then have, with $g_i := g(x_i)$

$$g(x) = (1 - r)g_i + rg_{i+1} \tag{27.216}$$

If the variance of the observed values is homogeneous, hence $\mathbb{V}(\underline{g}_i) = \sigma_n^2$, then the variance of the interpolated value is

$$\sigma_{\overline{n}}^2(x) = ((1 - r)^2 + r^2)\sigma_n^2 = (1 - 2r + 2r^2)\sigma_n^2 \quad \text{with} \quad r \in [0, 1] \,. \tag{27.217}$$

Hence, assuming homogeneous noise variance, we obtain the variance due to linear interpolation

$$\sigma_{\overline{n}}^2(r) = q_{\text{linear}}^2(r)\ \sigma_n^2 \tag{27.218}$$

with

$$q_{\text{linear}}^2(r) = 1 - 2r + 2r^2 \,. \tag{27.219}$$

The minimum is achieved for $r = 1/2$, namely $\sigma_{\overline{n}}^2(i + 1/2) = 1/2\ \sigma_n^2$.

The average variance is given by

$$\overline{\sigma_{\overline{n}}^2} = \int_{r=0}^{1} \sigma_x^2 \mathrm{d}r = \frac{2}{3}\sigma_n^2 \,. \tag{27.220}$$

The *correlation* can easily be determined if if the value of g is determined in neighbouring intervals, say $x = x_i + r$ and $y = x_i - 1 + s$, with $r, s \in [0, 1]$. Then we have

$$g(y) = (1 - s)g_{i-1} + sg_i \,. \tag{27.221}$$

Hence the correlation of the interpolated values is

$$\rho_{xy} = \frac{s(1 - r)}{\sqrt{((1 - r)^2 + r^2)((1 - s)^2 + s^2)}} \,. \tag{27.222}$$

It depends on both values x and y. For $r = s$, i.e., $x = y + 1$ we obtain

$$\rho_{x,x+1} = \frac{r(1 - r)}{1 - 2r + 2r^2} \in [0, 1/2] \,. \tag{27.223}$$

The maximum correlation is achieved if $r = s = 1/2$, as to be expected. The maximum correlation is 50%. For $s = 1 - r$ we obtain

$$\rho_{x,x+1} = \frac{(1 - r)^2}{1 - 2r + 2r^2} \in [0, 1] \,, \tag{27.224}$$

where the minimum is achieved for $r = s = 0$ and the minimum $\rho = 0$ is achieved for $r = s = 1$.

Therefore, we may derive individual variance depending on the interpolation point x.

27.7.1.2 2D linear interpolation

If the signal is given by the elements b_{nm} we have the function values $f(m, n) = b_{mn}$. The bilinear interpolation refers to the integer coordinates

$$i = \lfloor x \rfloor \quad \text{and} \quad j = \lfloor y \rfloor \,, \tag{27.225}$$

and depends on the coordinates

$$u = u(x) = x - i \quad \text{and} \quad v = v(y) = y - j\,. \tag{27.226}$$

and with the substitutions

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ u \end{bmatrix},\ \boldsymbol{v} = \begin{bmatrix} 1 \\ v \end{bmatrix},\ M_0 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},\quad \text{and} \quad B = \begin{bmatrix} b_{i,j} & b_{i,j+1} \\ b_{i+1,j} & b_{i+1,j+1} \end{bmatrix} \tag{27.227}$$

compactly can be written as

$$\boxed{f(x,y) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 B M_0^{\mathsf{T}} \boldsymbol{v}(y)\,.} \tag{27.228}$$

The variance of interpolated values $g(x,y)$ at $[x,y]$ uses the remainders $[r,s]$ and is given by

$$\sigma_{\overline{n}}(x,y) = (1 - 2r + 2r^2)(1 - 2s + 2s^2)\ ,\sigma_n^2\,, \tag{27.229}$$

see (27.142).

27.7.2 Cubic interpolation

27.7.2.1 1D cubic interpolation

Compact representation. Here we follow Xiao Shu, Bicubic interpolation, March 2013[2]. For each value x in the interval $[i, i+1]$ the function is a cubic polynomial which satisfies the following conditions

1. The function at i has the values $f(i) = b_i$
2. The function at $i+1$ has the value $f(i+1) = b_{i+1}$.
3. The derivative at i is $f'(i) = (b_{i+1} - b_{i-1})/2$.
4. The derivative at $i+1$ is $f'(i) = (b_{i+2} - b_i)/2$.

Hence, we need the four neighbouring values collected in

$$\boldsymbol{b} = \begin{bmatrix} b_{i-1} \\ b_i \\ b_{i+1} \\ b_{i+2} \end{bmatrix}. \tag{27.230}$$

We use the substitutions

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ (u-i) \\ (u-i)^2 \\ (u-i)^3 \end{bmatrix} \quad \text{and} \quad M_0 = \frac{1}{2}\begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \tag{27.231}$$

Then the interpolated value is

$$f(x) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 \boldsymbol{b}\,, \tag{27.232}$$

as above.

Proof: We use

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{27.233}$$

Then cubic function in the i-th interval $[i, i+1]$ can be written as

$$f^{(i)}(u) = a_0 + a_1(u-i) + a_2(u-i)^2 + a_3(u-i)^3 = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{a} \tag{27.234}$$

[2] https://www.ece.mcmaster.ca/~xwu/3sk3/interpolation.pdf

The derivative is

$$f_u^{(i)}(u) = a_1 + 2a_2(u-i) + 3a_3(u-i)^2 = \boldsymbol{u} \begin{bmatrix} a_1 \\ 2a_2 \\ 3a_3^2 \\ 0 \end{bmatrix} = \boldsymbol{u}^\mathsf{T} D \boldsymbol{a} \tag{27.235}$$

with the differentiation matrix

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{27.236}$$

The four conditions then can be written as

$$\begin{bmatrix} f^{(i)}(i) \\ f^{(i)}(i+1) \\ f_u^{(i)}(i) \\ f_u^{(i)}(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & -1/2 & 0 & 1/2 \end{bmatrix}}_{U} \begin{bmatrix} b_{i-1} \\ b_i \\ b_{i+1} \\ b_{i+2} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix}}_{V} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{27.237}$$

or compactly

$$\boldsymbol{c} = U\boldsymbol{b} = V\boldsymbol{a}\,. \tag{27.238}$$

Therefore

$$\boldsymbol{a} = V^{-1}U = M_0\boldsymbol{b}\,, \tag{27.239}$$

which holds since

$$U = VM_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & -1/2 & 0 & 1/2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \tag{27.240}$$

◇

Remark: This definition of interpolating cubic splines does not minimize the total curvature of the interpolating function, thus differs from the classical definition. In contrast to the classical definition, the function values $f(u)$ only depend on four neighbouring points b_i linearly, not on all values of the profile. ◇

Since, with $\boldsymbol{a} = M_0\boldsymbol{b}$ the first derivative of the polynomial from (27.235) we obtain the compact expression for the derivative

$$\boxed{f'(x) = \boldsymbol{u}^\mathsf{T}(x) M_1 \boldsymbol{b}} \tag{27.241}$$

with

$$M_1 = DM = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 & 0 \\ 4 & -10 & 8 & -2 \\ -3 & 9 & -9 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{27.242}$$

Variance of cubic interpolation. We now give the variances of cubic interpolation. Cubic interpolation at $r \in [0,1]$ requires the vales of g at $[-1,0,1,2]$. Specifically, we obtain the interpolated value

$$\begin{aligned} g(r) \quad = \quad & \frac{1}{2}\left[\left(-r + 2\,r^2 - r^3\right) g_{-1}\right. && (27.243) \\ & + \left(2 - 5\,r^2 + 3\,r^3\right) g_0 && (27.244) \\ & + \left(r + 4\,r^2 - 3\,r^3\right) g_1 && (27.245) \\ & \left. + \left(-r^2 + r^3\right) g_2\right], && (27.246) \end{aligned}$$

see (28.86). Assuming *homogeneous noise variance*, we obtain the variance

$$\sigma_{\overline{n}}^2(r) \quad = \quad q_{\text{cubic}}^2(r)\sigma_n^2 \tag{27.247}$$

with

$$\text{with} \quad q^2_{\text{cubic}}(r) = \frac{1}{2}\left[1 - 9\,r^2 + 8\,r^3 + 21\,r^4 - 30\,r^5 + 10\,r^6\right] . \tag{27.248}$$

It is symmetric w.r.t. $r = 1/2$. It reaches its maximum $\sigma_x^2(0) = \sigma_n^2$ at $r = 0$ and $r = 1$ and its minimum at $r = 1/2$ min

$$\sigma^2_{\overline{n}}(r = 1/2) = \frac{41}{64}\sigma_n^2 \approx 0.641\sigma_n^2 . \tag{27.249}$$

Observe, that this is only the effect of noise onto the interpolation, assuming a quite flat function f.

As in the case of linear interpolation, we may derive an individual variance as a function of the remainder $r = x - \lfloor x \rfloor$. We also can use the average variance, which is

$$\overline{\sigma^2_{\overline{n}}} = \int_{r=0}^{1} \sigma_x^2 \mathrm{d}r = \frac{57}{70}\sigma_n^2 \approx 0.814\sigma_n^2 . \tag{27.250}$$

On an average we then have an error of approximately 13%.

27.7.2.2 2D cubic interpolation

We thus obtain *bi-cubic interpolation* using the substitution

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{27.251}$$

and the collection of the 4×4 neighbouring values in the cell $[i, i+1] \times [j, j+1]$

$$B = \begin{bmatrix} b_{i-1,j-1} & b_{i-1,j} & b_{i-1,j+1} & b_{i-1,j+2} \\ b_{i,j-1} & b_{i,j} & b_{i,j+1} & b_{i,j+2} \\ b_{i+1,j-1} & b_{i+1,j} & b_{i+1,j+1} & b_{i+1,j+2} \\ b_{i+2,j-1} & b_{i+2,j} & b_{i+2,j+1} & b_{i+2,j+2} \end{bmatrix} , \tag{27.252}$$

We obtain

$$f(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 B M_0^{\mathsf{T}} \boldsymbol{v}(y) . \tag{27.253}$$

The partial derivatives then are

$$\boxed{f_x(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_1 B M_0^{\mathsf{T}} \boldsymbol{v}(y) \quad \text{and} \quad f_y(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 B M_1^{\mathsf{T}} \boldsymbol{v}(y)} \tag{27.254}$$

The variance of bi-cubic interpolated noise values is

$$\sigma^2_{\overline{n}}(x, y) = q^2_{\text{bi-cubic}}(r, s)\sigma_n^2 \quad \text{with} \quad q^2_{\text{bi-cubic}}(r, s) = q^2_{\text{cubic}}(r) q^2_{\text{cubic}}(s) . \tag{27.255}$$

The average variance is

$$\overline{\sigma^2_{\overline{n}}} = \int_{r=0}^{1} \int_{s=0}^{1} \sigma^2_{\overline{n}}(x, y) \mathrm{d}x\mathrm{d}y = \frac{57}{70}^2 \sigma_n^2 \approx 0.663\sigma_n^2 . \tag{27.256}$$

Interpolation error. We want to determine the interpolation error of a function $f(x, y)$ by

1. Interpolating the function f at the grid at $[i + 1/2, j + 1/2]$:

$$g(x, y) = f_B(x + 1/2, y + 1/2) . \tag{27.257}$$

2. Interpolating the the function g at the grid at $[i - 1/2, j - 1/2]$:

$$h(x, y) = g_B(x - 1/2, y - 1/2) . \tag{27.258}$$

3. Determining the error induced by the two interpolations

$$\sigma^2 = \mathbb{D}(h(x,y) - f(x,y))\,. \tag{27.259}$$

We start from (27.253) using

$$u = u(x) - i \quad \text{and} \quad v = v(x) - j\,, \tag{27.260}$$

with the special choice for $x = +1/2$

$$u_+ = x - \lfloor x \rfloor = +1/2 - 0 = 1/2 \quad \text{and} \quad v_+ = 1/2\,. \tag{27.261}$$

Hence, we have

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ 1/2 \\ 1/4 \\ 1/8 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v} = \begin{bmatrix} 1 \\ 1/2 \\ 1/4 \\ 1/8 \end{bmatrix} \tag{27.262}$$

We refer to the 49 values of $F(1:7,1:7)$, see Fig. 27.4

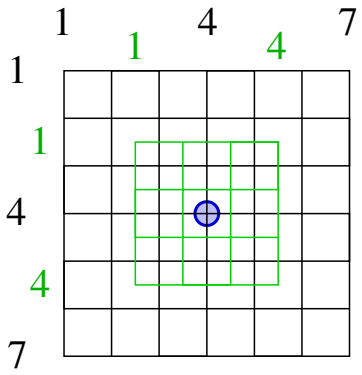


Figure 27.4: Interpolation error. The black 7×7 grid of f is used to interpolate the green 4×4 grid of g. This is used to obtain the interpolated value h at the position of $f(4,4)$. The difference is an indication for double the interpolation error.

This allows to derive $G(1:4,1:4)$ via

$$g(i,j) = \boldsymbol{u}^\mathsf{T} M_0 F(i:i+3, j:j+3) M_0^\mathsf{T} \boldsymbol{v}\,. \tag{27.263}$$

Similarly, starting from the interpolated signal g, we can derive the back shifted value

$$h = \boldsymbol{u}^\mathsf{T} M_0 G M_0^\mathsf{T} \boldsymbol{v}\,. \tag{27.264}$$

The difference

$$\delta(F) = h(F) - f(4,4) \tag{27.265}$$

is a function of the 49 values of F. The Jacobian $J = \partial\delta/\partial F$ is given by

$$J = \frac{1}{2^{16}} \begin{bmatrix} 1 & -18 & 63 & 164 & 63 & -18 & 1 \\ -18 & 324 & -1134 & -2952 & -1134 & 324 & -18 \\ 63 & -1134 & 3969 & 10332 & 3969 & -1134 & 63 \\ 164 & -2952 & 10332 & -38640 & 10332 & -2952 & 164 \\ 63 & -1134 & 3969 & 10332 & 3969 & -1134 & 63 \\ -18 & 324 & -1134 & -2952 & -1134 & 324 & -18 \\ 1 & -18 & 63 & 164 & 63 & -18 & 1 \end{bmatrix} \tag{27.266}$$

We now assume that the signal values f are correlated, depending on the underlying power-spectrum. We assume three different cases

1. Gaussian power spectrum. Then the covariance function is

$$C(d) = \exp(-(d/d_0)^2/2)\,. \tag{27.267}$$

2. Laplacian power spectrum. Then the covariance function is

$$C(d) = \frac{1}{1+(d/d_0)^2}\,. \tag{27.268}$$

3. The powers spectrum follows the power law $P(u) \propto f^{-2}$, or $P(u) = 1/(1+u^2)/\pi$. Then the covariance function has the form

$$C(d) = \exp(-d/d_0)\,. \tag{27.269}$$

The value d_0 controls the smoothness of the signal, larger d_0 leads to smoother functions. Via variance propagation we obtain $\mathbb{V}(\delta(f))$. Since this difference results from two interpolation we report $\sigma_\delta = \sqrt{\mathbb{V}(\delta)/2}$ in the table. We observe: (1) the larger d_0, i.e., the

d_0	P(Gauss)	P(Lapl)	P(Power)
1.0000	0.1855	0.2888	0.3499
1.5000	0.0643	0.1860	0.2947
2.0000	0.0249	0.1180	0.2583
2.5000	0.0112	0.0754	0.2323
3.0000	0.0056	0.0491	0.2128
3.5000	0.0031	0.0327	0.1974
4.0000	0.0019	0.0222	0.1849

Table 27.1: Standard deviation of bi-cubic interpolation error

smoother the signal, the smaller the interpolation error is. (2) As to be expected, signals with Laplacian power spectrum are rougher than those with Gaussian, and smoother than those with thepower spectrum following the power law.

These results refer to the shift $[1/2, 1/2]$ and depend on the assumed stochastical model for the signal. For the use in LSM, we do not want a dependency on the stochastical model for the unknown image function. Since $\mathsf{J} = \partial\delta/\partial\mathsf{F}$, we can use the matrix J as filter for deriving δ from f:

$$\delta = \mathsf{J} * f\,. \tag{27.270}$$

Obviously, the matrix J represents a highpass filter. Hence, we can determine the individual interpolation errors due to forward and backward shifting with interpolation from any signal. This allows us to derive the variance of the maximal interpolation error from

$$\widehat{\sigma}^2_{\delta,\max} = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \delta_r^2\,. \tag{27.271}$$

Since we are interested in the average interpolation error within the region of one pixel, we determine the mean of the expected variance $\overline{\sigma^2_\delta}$ for all forward and backward shifts $[x, y], x, y \in [0, 1]$. Hence, we can determine the interpolation variance for a specific window f from

$$\widehat{\overline{\sigma^2_\delta}} = k\,\widehat{\sigma}^2_{\delta,\max} \quad \text{with} \quad k = \frac{\overline{\sigma^2_\delta}}{\sigma^2_{\delta,\max}}\,. \tag{27.272}$$

As an example we have the variances of the interpolation error for $d_0 = 1$ and Gaussian

covariance function at the grid points $[i,j]/8, i,j \in \{1,...,9\}$ of the unit interval:

$$[\sigma_{ij}^2] = \frac{1}{1000}\begin{bmatrix} 0 & 9 & 93 & 247 & 327 & 247 & 93 & 9 & 0 \\ 9 & 22 & 111 & 267 & 349 & 267 & 111 & 22 & 9 \\ 93 & 111 & 208 & 368 & 449 & 368 & 208 & 111 & 93 \\ 247 & 267 & 368 & 527 & 608 & 527 & 368 & 267 & 247 \\ 327 & 349 & 449 & 608 & 688 & 608 & 449 & 349 & 327 \\ 247 & 267 & 368 & 527 & 608 & 527 & 368 & 267 & 247 \\ 93 & 111 & 208 & 368 & 449 & 368 & 208 & 111 & 93 \\ 9 & 22 & 111 & 267 & 349 & 267 & 111 & 22 & 9 \\ 0 & 9 & 93 & 247 & 327 & 247 & 93 & 9 & 0 \end{bmatrix} \tag{27.273}$$

As to be expected, the interpolation error is zero in the corners of the square, thus for integer coordinates.

Interestingly, the factor k does not vary much for different stochastical models of f, as Tab. 27.2 shows. Therefore, we can use the following variances of the model error of the

d_0	P(Gauss)	P(Lapl)	P(Power)
1.0000	0.3482	0.3710	0.3803
1.5000	0.3425	0.3608	0.3793
2.0000	0.3432	0.3536	0.3789
2.5000	0.3446	0.3490	0.3787
3.0000	0.3459	0.3462	0.3786
3.5000	0.3468	0.3445	0.3785
4.0000	0.3476	0.3437	0.3785

Table 27.2: Ratio k of mean and maximal bicubic interpolation error variance for different stochastical models for a signal.

given data, e.g., for g in grey levels $[0,...,255]$

$$\sigma_{n'_j}^2 = \sigma_{n_j}^2 + k\,\hat{\mathbb{V}}(J * f) \quad \text{with} \quad k = 0.37\,. \tag{27.274}$$

depending on the empirical noise variance $\sigma_{n_j}^2$ of the intensities of g_i and the interpolation error derived from the image window f.

27.7.3 Quadratic smoothing

27.7.3.1 1D quadratic smoothing

Here, the quadratic function $f(x)$ is valid for points x in the interval $[i-1/2, i+1/2]$. The slopes at the boundary points are

$$f'(i-1/2) = b_i - b_{i-1} \quad \text{and} \quad f'(i+1/2) = b_{i+1} - b_i\,. \tag{27.275}$$

However, the function at i does not have the value b_i, but a smoothed value. Then with the substitutions

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ u \\ u^2 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_{i-1} \\ b_i \\ b_{i+1} \end{bmatrix}, \tag{27.276}$$

and

$$M_0 = \frac{1}{8}\begin{bmatrix} 1 & 6 & 1 \\ -4 & 0 & 4 \\ 4 & -8 & 4 \end{bmatrix}, M_1 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{27.277}$$

we obtain the interpolated value and the corresponding derivative as

$$f(x) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 \boldsymbol{b} \quad \text{and} \quad f'(x) = \boldsymbol{u}^{\mathsf{T}}(x) M_1 \boldsymbol{b} \tag{27.278}$$

27.7.4 Smoothing cubic B-splines

Cubic B-splines also do not pass through the given points (m, n, b_{mn}). But can be represented in the same way as cubic interpolation, see see http://ps-2.kev009.com/tl/techlib/manuals/adoclib/aixprggd/gl32prgd/drawwfcv.htm.

With the 4×4 values collected in the matrix B the vectors

$$\boldsymbol{u}(u) = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}, \qquad \boldsymbol{v}(v) = \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{27.279}$$

and the matrices

$$M_0 = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad \text{and} \quad M_1 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 & 0 \\ 2 & -4 & 2 & 0 \\ -1 & 3 & -3 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{27.280}$$

we obtain the surface and its derivatives by (27.253) and (27.254).

27.8 Experiments

Remark: The following experiments are incomplete. ◇

We assume we have the following routines, all minimizing the sum of squared residuals.

- TEMM = template matching:
  - the template is a known continuous function,
  - no interpolation is required.
- ALSM = asymmetric image matching:
  - using pixels in the observed template which are reachable in the observed image,
  - taking a mean gradient,
  - applying cubic interpolation.
- CLSM = classical asymmetric image matching:
  - using pixels in the observed template which are reachable in the observed image
  - taking the simple gradient with $[-1, 0, 1]/2$ only in the template
  - applying linear interpolation.
- SLSM = symmetric image matching:
  - using pixels in the observed images which map to a mean unknown image,
  - taking a mean gradient,
  - applying cubic interpolation.

We add the dimension d and the number of geometric and radiometric parameters in brackets. E.g. the symbol TEMM1(1,2) refers to 1D template matching with 1 geometric parameter (shift) and 2 radiometric parameters (affinity).

We did not realize models for all methods, see table

List of experiments/Questions:

- Is the implementation correct using the three tests?
- Are the results different, when using the generated real valued intensities or the rounded intensities?
- Demonstrate the need for changing the model variance due to bicubic interpolation.

method	1(1,2)	1(2,2)	2(4,2)
TEMM	x		
			x
ALSM	x	x	
			x
CLSM		x	
SLSM		x	
			x

Table 27.3: Realized methods

- Demonstrate the differences when using pyramids, especially the number of iterations (todo).
- What is the radius of convergence? How does it depend on the size of the windows, the texture and the use of pyramids (todo).
- Are the parameters of Lowe-points sufficient for convergence? How far are they apart? How is the distribution of the scale and direction information?
- How large is the difference to asymmetric image matching?
- How large is the difference to classical image matching?
- Do triplet-closure tests reveal differences? (todo)
- How large is the difference to using only pixels with gradient? (todo)

27.8.1 Verification of correctness

We first check the correctness of the implementation, i.e., the consistency between theory and implementation based on simulated data. We present results for 1D for all four methods. We report tests on repeated trials. For each test we specify

1. model
2. number of parameters
3. redundancy
4. number of trials
5. normalized outcome of statistical tests for
   (a) variance factor $\hat{\sigma}_0^2$,
   (b) covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$, and
   (c) bias $\boldsymbol{b} = \hat{\boldsymbol{\mu}}_{\hat{\theta}} - \tilde{\boldsymbol{\theta}}$

   We normalize the output such that the test statistic is in the range $[-1, +1]$. Hence, if the test statistic is T and the two-sided interval is $[l, h]$ we provide

$$p(T; l, h) = \frac{\log \frac{T}{\sqrt{lh}}}{\log \sqrt{\frac{h}{l}}} \quad \text{with} \quad p(l; l, h) = -1 \quad \text{and} \quad p(h; l, h) = +1\,. \tag{27.281}$$

   Hence, the test statistic is in the confidence interval if $p \in [-1, +1]$.
6. We give the tests for both, all parameters ($p(\cdot)$), only the geometric ones ($p_g(\cdot)$), and only the radiometric ones ($p_r(\cdot)$).

We generate the data with the following parameters

- signal with Gaussian spectrum with effective bandwidth $\sigma = b_{\text{eff}}$ and random phase from L frequencies with

$$\mu_a = 128\ [\text{gr}]\,, \quad \sigma_a = 40\ [\text{gr}]\,, \quad b_{\text{eff}} = 0.05\,, \quad L = 40\,. \tag{27.282}$$

- true parameters, if applicable:
  - 1D matching

$$s = 1.0\,, \qquad u = 4.6\,, \qquad a = 1.05\,, \qquad b = 10 \text{ [gr]} \tag{27.283}$$

    We choose $s = 1.0$ in order to be able to compare the solutions with 3 and 4 parameters.
- approximate values identical to true values.
- noise standard deviation $\sigma_n = 1$ [gr].
- starting seed for random generation: 1.

Table 27.4: Tests on the correctness of the estimations of 1D signals , the ideal template matching (TEMM) with and without rounding (r.) and asymmetric and symmetric matching, ALSM and SLSM, respectively. Number of geometric and radiometric unknowns (U_g, U_r), redundancy R, number of samples K, normalized test statistics p for variance factor, for covariance matrix $\boldsymbol{\Sigma}$, and bias b, for all and only geometric parameters p and p_g, respectively.

	dim	round	$U_g + U_r$	R	K	$p(\sigma_0)$	$p(\boldsymbol{\Sigma})$	$p(\boldsymbol{b})$	$p_g(\boldsymbol{\Sigma})$	$p_g(\boldsymbol{b})$
1	TEMM1	−	1+2	22	1000	0.3436	0.3280	0.4447	-	-
2	TEMM1	+	1+2	22	1000	0.6480	0.3102	0.2786	-	-
3	ALSM1	+	1+2	26	1000	-4.7756	0.4630	0.8395	0.4219	0.9374
4	ALSM1	+	2+2	25	1000	-4.7883	0.3571	0.7303	0.5696	0.8072
5	SLSM1	+	2+2	34	1000	1.0760	-0.5966	0.9015	0.0694	0.8826

We can draw the following conclusions from the table:

- There is no reason to doubt the estimated variance factor for template matching. Asymmetric least squares matching severely underestimates the variance factor, while symmetric least squares matching slightly overestimates the variance factor.
- In all cases the estimates appear to have no bias, referring to all parameters or referring only to the geometric parameters.
- In all cases, except one, the predicted covariance matrix of the estimated parameters appears to reflect their uncertainty. An exception is the underestimation of the covariance matrix for symmetric least squares matching (SLMS1, with $p(\boldsymbol{\Sigma}) = -0.6$). However, if we take the covariance matrix of the parameters corrected by the estimated variance factor, thus

$$\hat{\sigma}_0^2 \boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}} \tag{27.284}$$

  this corrected covariance matrix is acceptable, since the p-values in the table – due to their logarithmic character, see (27.281) – add.

Hence, in all cases we can rely on the estimates and on the internal prediction of their uncertainty. Only the estimated variance factor, i.e., the estimate of the noise variance appears unreliable for asymmetric least squares matching.

27.8.2 Convergence

The example shows 1D asymmetric matching ALSM1(1,2) with geometric shift and radiometric affinity, i.e., 3 unknown parameters. It shows the adaptation of the template to the observed signal. The true parameters are

$$u = -2.3\,, \quad a = 1.0432\,, \quad b = 1.8841/256 \approx 0.0074\,. \tag{27.285}$$

The brightness parameter b refers to the signal range $[0, 1]$ and corresponds to 30 [gr]. The noise is assumed to be

$$\sigma_g = \sigma_h = 2/256\,. \tag{27.286}$$

We start from approximate values

$$u = -5.9\,, \quad a = 1.4432\,, \quad b = -0.1098\,. \tag{27.287}$$

Figure 27.6 shows how the signal g (blue) is adapting to the signal h (red). Again, due to the large deviation between approximate and true values 5 iterations are necessary, to fulfil the convergence criterium. However, visually already 3 iterations appear sufficient.

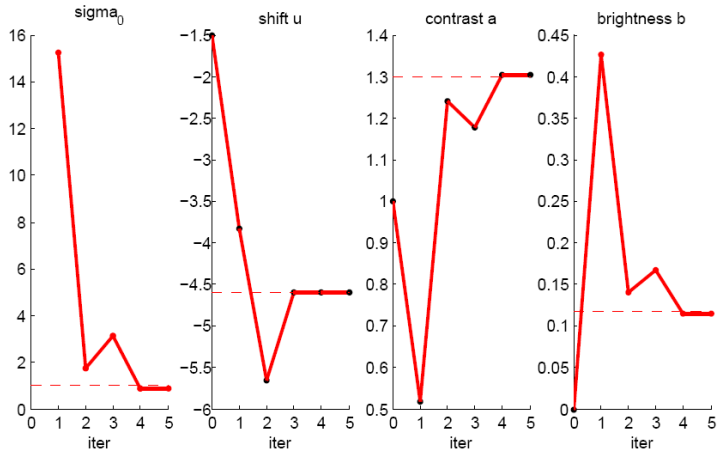


Figure 27.5: Example for template matching TEMM1(1,2). Convergence of parameters

ν	$\Delta\hat{u}$	$\Delta\hat{a}$	$\Delta\hat{b}$	N	$\hat{u}$	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_0$
0					-5.9000	1.4432	-0.1098	
1	4.2926	-0.8707	0.3242	28	-1.6074	0.5725	0.2144	3.8060
2	-0.6618	0.3259	-0.1429	32	-2.2692	0.8984	0.0716	0.9904
3	-0.0330	0.1091	-0.0471	31	-2.3022	1.0076	0.0245	1.0865
4	0.0012	0.0069	-0.0029	31	-2.3010	1.0144	0.0215	1.1062
5	*0.0000	*0.0000	*0.0000	31	-2.3010	1.0145	0.0215	1.1064
$\tilde{\boldsymbol{\theta}}$					-2.3000	1.0432	0.0074	
$\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}$					-0.0010	0.0237	0.0141	
σ					0.0418	0.0165	0.0081	

Table 27.5: Classical 1D-LSM, i.e., TEMM1(1,2), with geometric shift and two radiometric parameters. Parameters as a function of the iteration ν. Observe, the number N of observations changes, depending on the estimated shift u. The differences between the estimated parameters $\hat{\boldsymbol{\theta}}$ and the true parameters $\tilde{\boldsymbol{\theta}}$ is in the range of the standard deviations. * The 0-values are less than 10^{-5}

27.8.3 Examples

27.8.3.1 1D Template matching

The example shows 1D template matching TEMM1(1,2) with geometric shift and radiometric affinity, i.e., 3 unknown parameters. It shows the adaptation of the template to the observed signal. The true parameters are

$$u = -4.6\,, \quad a = 1.3\,, \quad b = 30/256 \approx 0.1172\,. \tag{27.288}$$

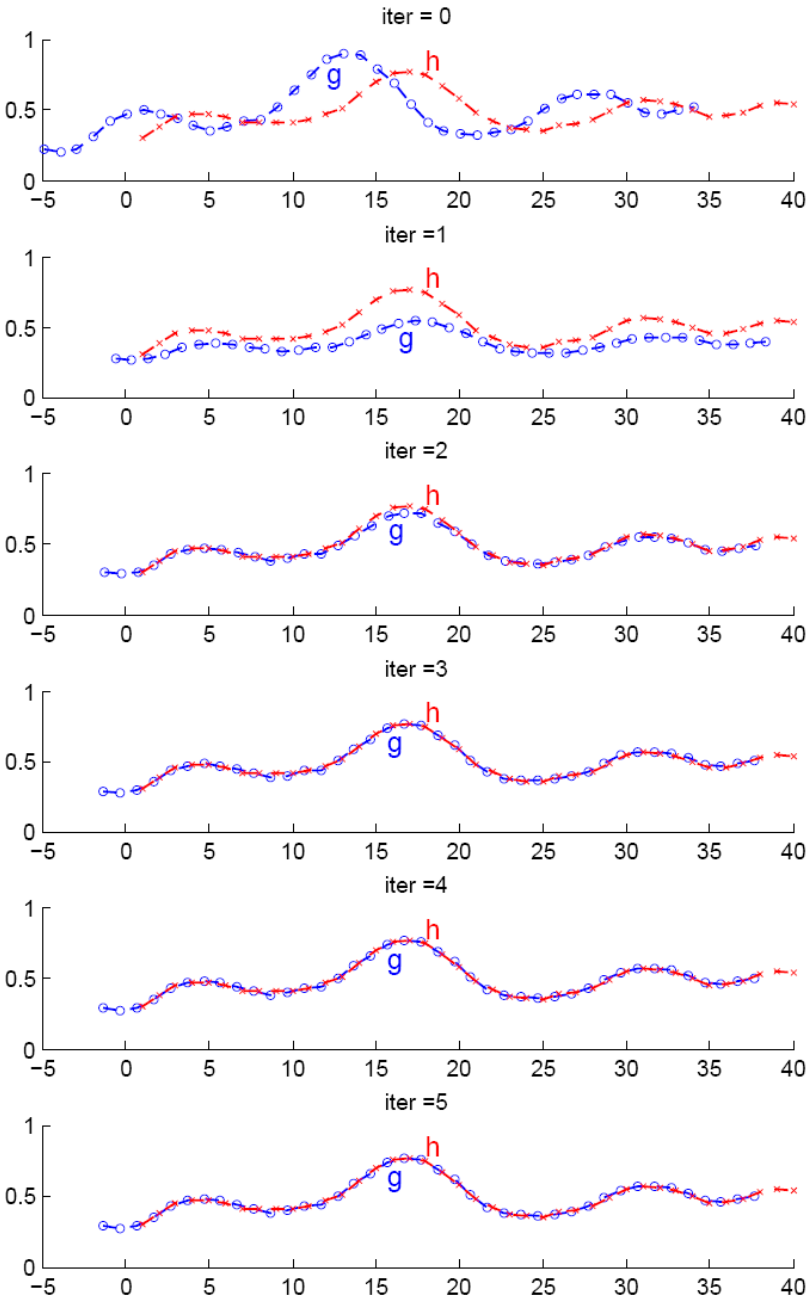


Figure 27.6: Asymmetric LSM, ALSM1(1,2), with geometric shift and two radiometric parameters. Evolution of adaptation of g (blue) w.r.t. h (red)

The brightness parameter b refers to the signal range $[0, 1]$ and corresponds to 30 [gr]. The noise is assumed to be

$$\sigma_n = 1/256\,. \tag{27.289}$$

We start from approximate values

$$u = -1.6\,, \quad a = 1.0\,, \quad b = 0\,. \tag{27.290}$$

Fig. 27.7 shows how the template (red) is adapting to the observed signal (blue). Due to the large deviation between approximate and true values 5 iterations are necessary. Iterations are terminated if the change of a parameter is less than 10% of its standard

deviation. Table 27.6 shows the change of the parameters and of the estimated σ_0. This is visualized in Fig. 27.5.

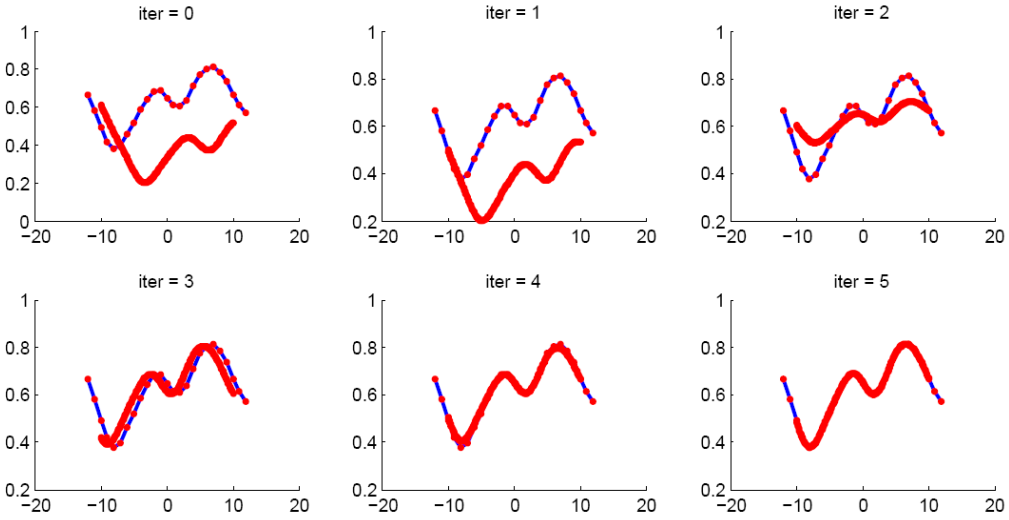


Figure 27.7: Template matching TEMM1(1,2). Adaptation of f (red) to g (blue) over five iterations

ν	$\Delta\hat{u}$	$\Delta\hat{a}$	$\Delta\hat{b}$	$\hat{u}$	$\hat{a}$	$\hat{b}$	$\hat{\sigma}_0$
0				−1.5000	1.0000	0.0000	
1	−2.3308	−0.4809	0.4257	−3.8308	0.5191	0.4257	15.245
2	−1.8167	0.7214	−0.2851	−5.6476	1.2405	0.1407	1.745
3	1.0527	−0.0642	0.0265	−4.5948	1.1763	0.1672	3.155
4	−0.0023	0.1274	−0.0517	−4.5972	1.3037	0.1154	0.871
5	0.0002	*0.0000	*0.0000	−4.5970	1.3037	0.1154	0.871
$\tilde{\boldsymbol{\theta}}$				−4.6000	1.3000	0.1172	
$\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}$				−0.0030	0.0037	−0.0018	
σ				0.0153	0.0088	0.0035	

Table 27.6: Template matching TEMM1(1,2). Parameters as a function of the iteration ν. The differences between the estimated parameters $\hat{\boldsymbol{\theta}}$ and the true parameters $\tilde{\boldsymbol{\theta}}$ is in the range of the standard deviations. * The 0-values are less than 10^{-7}

27.8.3.2 1D LSM methods with shift and scale

We now compare the effect of exchanging the two signals onto the estimated parameters for three methods

1. Symmetric LSM (SLSM1): expecting zero influence
2. Asymmetric LSM (ALSM1): expecting small influence due to the rigour of all steps, and
3. Classical LSM (CLSM1): expecting larger influences, due to the simplicity of the observed intensity differences and the gradient computation.

In all cases we apply the following procedure

- the model contains four parameters, two geometric and two radiometric affinities.
- use the same type of signal as above (mean, standard deviation and roughness)

- perform $K = 1000$ repetitions with random variation of the shift in the range $[-0.5, +0.5]$ [pel] and varying noise.
- Performing the three checks on the correctness of the implementation.
- Determining the differences d_u

$$d_u = p_{u,\text{backward}} - p_{u,\text{forward}} \tag{27.291}$$

of the four parameters $p_u \in \{s, u; a, b\}$, characterizing them by their normalized mean and standard deviation

$$m = \frac{\widehat{d_u}}{\sigma_{p_u}} \quad \text{and} \quad s = \frac{\widehat{\sigma}_{d_u}}{\sigma_{p_u}}\,, \quad u = 1, 2, 3, 4\,, \tag{27.292}$$

and providing a histogram.

- to some degree vary the noise variance

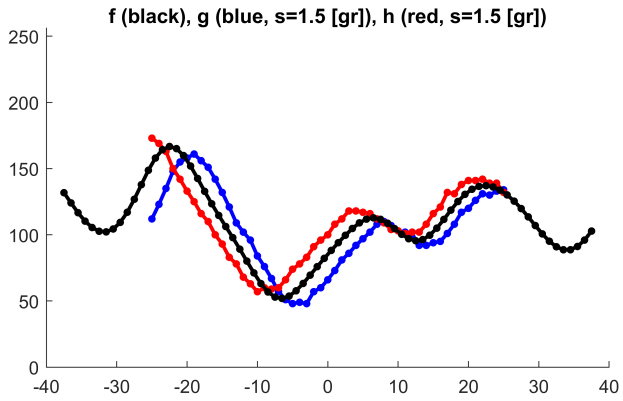


Figure 27.8: Sample. Case $\sigma_n = 1$ [gr], geometric scale $s = 1.1$

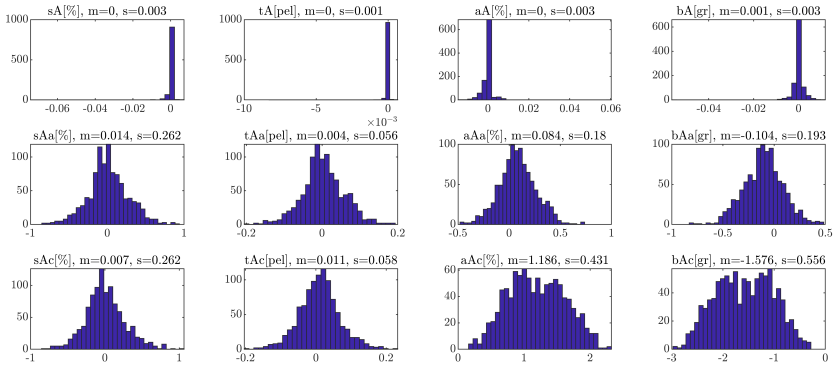


Figure 27.9: Effect. Case $\sigma_n = 1$ [gr], , geometric scale $s = 1.1$

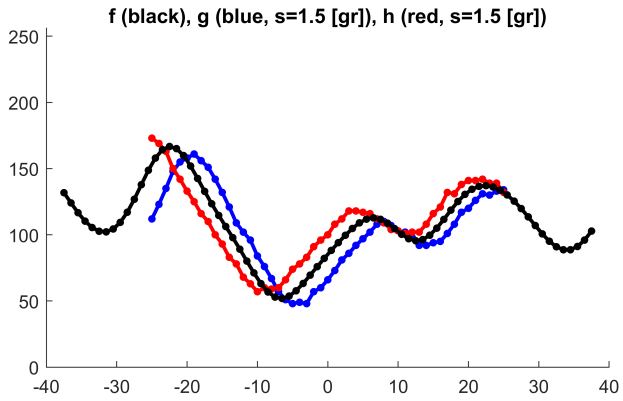


Figure 27.10: Sample for the true (black) and the observed signals (blue and red). Case $\sigma_n = 1.5$ [gr], geometric scale $s = 1.1$

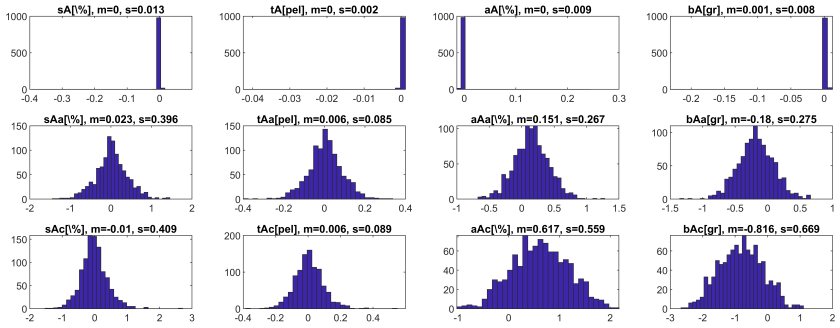


Figure 27.11: Effect of exchanging the two signals on the four parameters. Case $\sigma_n = 1.5$, , geometric scale $s = 1.1$

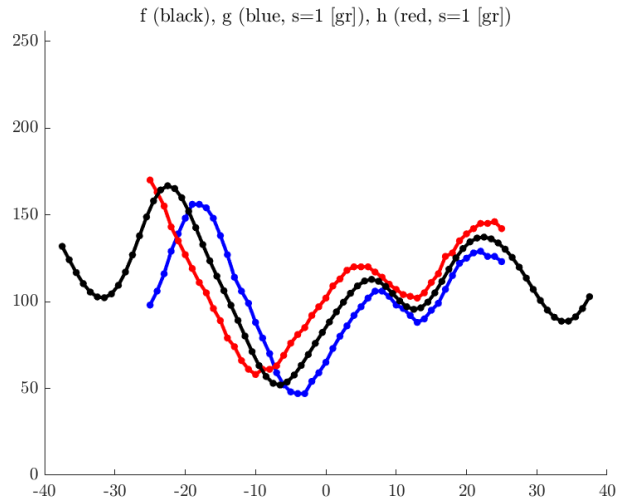


Figure 27.12: Effect. Case $\sigma_n = 1$, geometric scale $s = 1.25$

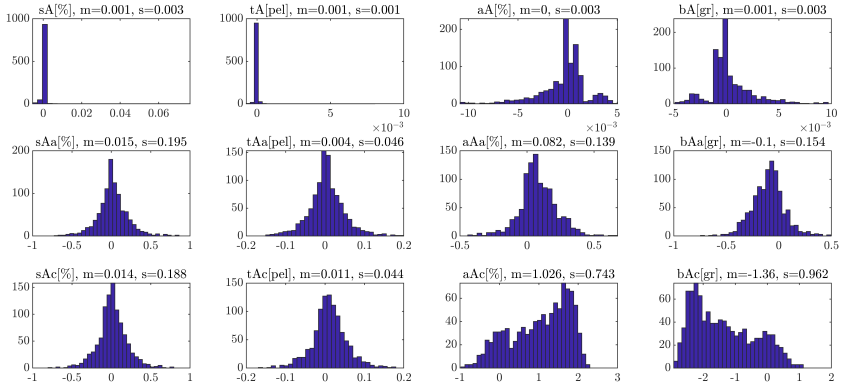


Figure 27.13: Effect. Case $\sigma_n = 1$ [gr], geoemtric scale $s = 1$

27.8.3.3 1D symmetric LSM with shift and scale

tbd.

27.8.3.4 2D asymmetric LSM with shift only

tbd.

27.8.3.5 2D symmetric LSM with affinity

tbd

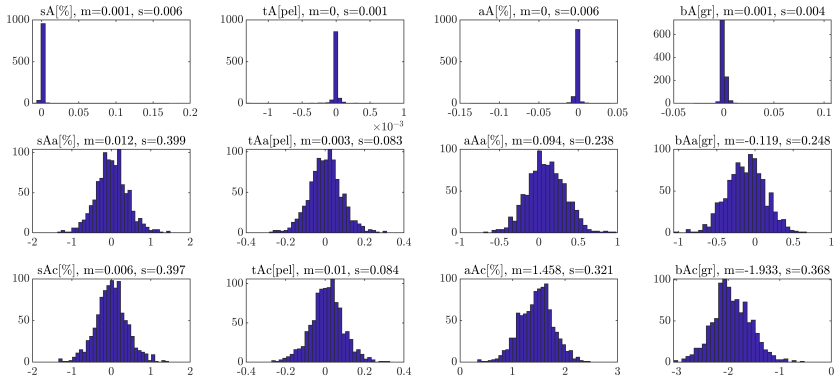


Figure 27.14: Effect. Case $\sigma_n = 1$ [gr], geometric scale $s = 1.25$

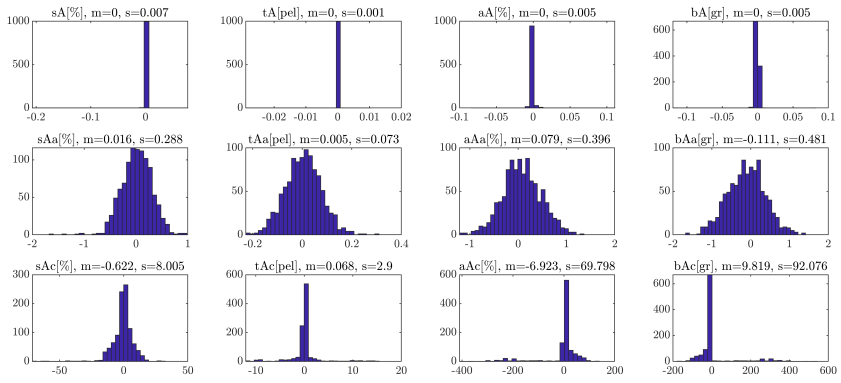


Figure 27.15: Effect. Case $\sigma_n = 1$, geometric scale $s = 1.5$

28 Symmetric Least Squares Matching — Sym-LSM

The note is a report which describes the basics of symmetric least squares matching (Sym-LSM) which is useful for high-precision image matching and realized in Matlab.

28.1 Summary

The report describes the basics of symmetric least squares matching (Sym-LSM) which is useful for highprecision image matching and realized in Matlab.

The principle of Sym-LSM is to minimize the weighted sum of the squares of the residuals of the intensity of two images g and h and this way obtains statistically optimal estimates for the parametrized geometric and radiometric distortions between two overlapping images. The estimated geometric transformation may be used in the context of relative image orientation for refining the coordinates of corresponding keypoints or for tracking keypoints within a video sequence. We assume images are grey-level images.

The main properties of the method are the following:

1. In contrast to classical approaches, exchanging the two signals leads to identical results, i.e., mutually inverse geometric and radiometric transformations.

2. LSM can be realized such that – for sets of simulated data – the three statistical tests on the correctness of the implementation generally do not fire:
   (a) The estimated variance factor does not differ from 1 too much. This indicates that model and (simulated) data are consistent.
   (b) The empirical covariance matrix derived from samples does not significantly differ from the theoretical covariance matrix, which is the Cramer-Rao bound. This suggests, that the theoretical covariance matrix can be used as reliable uncertainty indicator.
   (c) The estimates show no significant bias.

   Therefore, we can rely on the estimated parameters and their covariance matrix in real cases if convergence is achieved and the variance factor does not differ from 1 too much. If $\widehat{\sigma}_0$ is not close to 1, this indicates, the assumed model does not fit to the observations. The cause of this effect cannot be given: it may be, that the scene is not flat, the windows are too large, the estimated noise variance deviates from the noise variance in the windows, the shadow situation in both images is different, and so on.
3. Individual variances for all observations can be taken into account, especially, position or signal depending variances. Covariances are neglected. The signal dependent noise variances can be derived from given images without needing any control parameters.
4. Interpolation of the observed signals is necessary for reconstructing the unknown signal. This causes some – generally negligible – approximations in the method.
5. For affine geometric transformations the similarity transformation derivable from corresponding Lowe-keypoints can be used as approximate transformation.
6. From the estimated parameters and their covariance matrix one can derive refined point correspondences of affine correspondences. The affine parameters on an average have a relative accuracy a few percent, and the estimated shifts/parallaxes a standard deviation below 0.1 pixels.
   The affine correspondences can be used for estimating the relative pose of two calibrated or partially calibrated cameras, i.e., with or without focal length based on pairs of affine matches.

The MATLAB software provided consists of a core routine

`LSM_62_sym_main.m`

and several demo routines

- `demo_LSM_small.m` for showing the use of the main routine,
- `demo_LSM_medium.m` for showing the use of keypoints and the noise variance estimation,
- `demo_LSM_simulated.m` for checking the implementation with simulated data, and
- `demo_LSM_image_pairs.m` for manually providing keypoints in real data.

On Notation. Signals are two-dimensional function. The function names taken from the middle of the alphabet, e.g., f, g, and h. Coordinate names are taken from the end of the alphabet: so e.g., $f(\boldsymbol{x})$, $g(\boldsymbol{y})$, and $h(\boldsymbol{z})$. Sometimes, we use the convention $\boldsymbol{x} = [x, y]^\mathsf{T}$. Functions depend on coordinates $\boldsymbol{x}_i$, where the index range is a set of integers $\in \mathbb{Z}$ If a coordinate $\boldsymbol{x}$ has no index, the context tells whether it is a real or an integer. If two coordinates, say $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ have the same index they refer to corresponding points. Homogeneous coordinates and matrices are boldface upright, e.g., coordinates $\mathbf{x}$ or transformation A. The dimension depends on the context. Stochastical variables are underscored, e.g., the discrete noise function is $\underline{m}(\boldsymbol{x}_i)$. Sets and names are written with calligraphic letters, e.g., the set of all pixels in the first image is $\mathcal{g} = \{(x, y, g)_i, i = 1, ..., I\}$.

We denote the two images g and h as left and right image or first and second image, depending on the context.

28.2 Image Matching

This section describes the details of the model underlying the matching approach and the method for estimating the unknown parameters and their uncertainty.

The pipeline for using the matching procedure SYM-LSM is shown in Fig. 28.1.

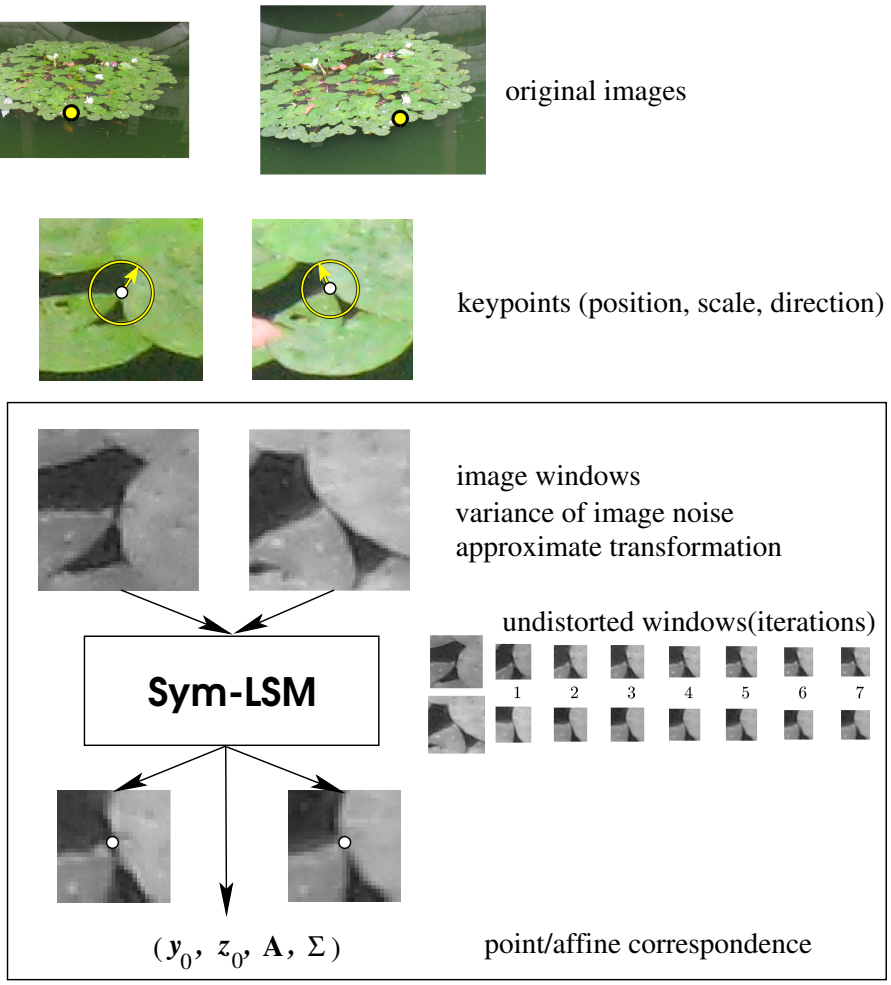


Figure 28.1: Pipeline for image matching. Starting from two images and measured keypoints with position, scale and direction, two corresponding windows are determined as input to the matching procedure SYM-LSM. Given approximate values and variances for the image noise it iteratively mutually undistorts the two image windows. After convergence it provides best estimates for the geometric and radiometric transformation between the two image windows, which allow to derive a point or an affine correspondence together with its covariance matrix

28.2.1 Model

Let the two square image windows $g(\boldsymbol{y})$ and $h(\boldsymbol{z})$ be given, see Fig. 28.2. The coordinates refer to the centre of the windows. We assume, both windows are noisy observations of an unknown true underlying signal $f(\boldsymbol{x})$, with individual geometric distortion, brightness, and contrast. We want to determine the geometric distortion $\boldsymbol{z} = \mathcal{A}(\boldsymbol{y})$ and the radiometric

distortion $h = \mathcal{R}(g) = pg + q$. Classical matching methods, assume the geometric and radiometric distortion of one of the two windows is zero, e.g., assuming the reference image is identical to the first image $g(\boldsymbol{y}) = f(\boldsymbol{x})$, with $\boldsymbol{y} = \boldsymbol{x}$.

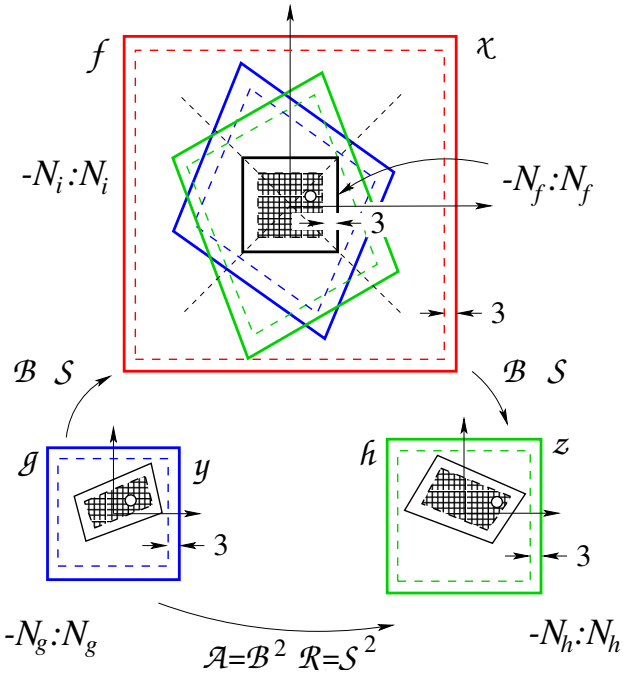


Figure 28.2: Relations between two given square image patches $g(\boldsymbol{y})$ (blue) and $h(\boldsymbol{h})$ (green) and the mean patch $f(\boldsymbol{x})$ (which is the black within the red region). The two image patches g and h are related by geometric and radiometric affinities $\mathcal{B}$ and $\mathcal{S}$, respectively. The correspondence is established by the patch f. Geometrically and radiometrically it lies in the middle between g and h. Only a region in the overlap of the two patches g and h mapped to f can be used. We choose the maximum square (black). The observations are all pixels in g and h which map into the black square of the reference image f. We assume the reference image f is a restored version of the weighted mean of the two projected images g and h. The patches g and h may have different sizes. The size of the unknown signal (black, textured) depends on the sizes of g and h, the approximate affine transformation $\mathcal{A} = \mathcal{B}^2$ and a border to allow bi-cubic interpolation, and is adapted in each iteration. The large image in the χ-frame is used for generating artificial images. The radiometric transformation $\mathcal{R} = \mathcal{S}^2$ is splitted in the same way (not shown here). The size of the windows is given by the ranges of the pixels, e.g., for image f the range is $-N_i : N_i := -N_i, ..., N_i$

We break this asymmetry by placing the unknown signal $f(\boldsymbol{x})$ in the middle between the observed signals between g and h:

$$g(\boldsymbol{y}) \overset{\mathcal{B},\mathcal{S}}{\longrightarrow} f(\boldsymbol{x}) \overset{\mathcal{B},\mathcal{S}}{\longrightarrow} h(\boldsymbol{z}) \quad \text{such that} \quad \mathcal{A} = \mathcal{B}^2 \,, \mathcal{R} = \mathcal{S}^2 \,. \tag{28.1}$$

The geometric and the radiometric transformations $\mathcal{A}$ and $\mathcal{R}$ are split into the sequence of two identical geometric and radiometric transformations $\mathcal{B}$ and $\mathcal{S}$, with the signal $f(\boldsymbol{x})$ having the same distortion w.r.t. $g(\boldsymbol{y})$ as the signal $h(\boldsymbol{z})$ has w.r.t. $f(\boldsymbol{x})$. Therefore the complete geometric transformations $\mathcal{A}$ and $\mathcal{R}$ result from two times applying $\mathcal{B}$ and $\mathcal{S}$ to $g(\boldsymbol{y})$: hence, we have $\boldsymbol{z} = \mathcal{B}(\mathcal{B}(\boldsymbol{y})) = \mathcal{B}^2(\boldsymbol{y})$ and $h = \mathcal{S}(\mathcal{S}(\boldsymbol{g})) = \mathcal{S}^2(g)$.

Assuming affinities for the geometric and the radiometric distortion, we have the following generative model. The geometric and the radiometric models for the two images are

$$\boldsymbol{y} \mapsto \boldsymbol{x}: \quad \boldsymbol{x} = B\boldsymbol{y} + \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x} \mapsto \boldsymbol{z}: \quad \boldsymbol{z} = B\boldsymbol{x} + \boldsymbol{b} \tag{28.2}$$

$$g \mapsto f: \quad f = sg + t \quad \text{and} \quad f \mapsto h: \quad h = sf + t \tag{28.3}$$

with

$$B = \begin{bmatrix} b_1 & b_3 \\ b_2 & b_4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} b_5 \\ b_6 \end{bmatrix} \tag{28.4}$$

In the following we collect the eight unknown parameters of the affinities in the vector

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_G \\ \boldsymbol{\theta}_I \end{bmatrix} \quad \text{with} \quad \boldsymbol{\theta}_G = [b_1, b_2, b_3, b_4, b_5, b_6]^\mathsf{T} \quad \text{and} \quad \boldsymbol{\theta}_I = [s, t]^\mathsf{T} . \tag{28.5}$$

Hence, we have the relation of the geometric transformations $\mathcal{B}$ to the compound transformation $\mathcal{A}$ with the six parameters in $\boldsymbol{a}$

$$\mathsf{A} = \begin{bmatrix} A & \boldsymbol{c} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \mathsf{B}^2 \quad \text{with} \quad A = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{c} = \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \tag{28.6}$$

This model is rigorous only, if

1. the scene surface is planar in a differentiable region, and
2. the intensity differences result from brightness or contrast changes only.

Hence, the geometric model is adequate, if the surface is smooth and the window size is not too large. Usually we have window sizes between 15×15 and 100×100 pixels, but larger windows may be fine in special cases. The radiometric model is adequate if there are no occlusions or local illumination differences, e.g., caused by shadows, when the images are not taken simultaneously.

We now assume the intensities g and h are noisy and distorted versions of an underlying true signal f with standard deviations $\sigma_n(g)$ and $\sigma_m(h)$ depending on g and h. Integrating the geometry and intensity transformation we arrive at the following model, which is generative, i.e., allows to simulate observed images. Using the radiometric transformations (28.3) we first obtain for all pixels j and k in the first and the second image and the corresponding pixels in the reference image f

$$\underline{g}(\boldsymbol{y}_j) = 1/s \cdot (f(\boldsymbol{x}_j) - t) + \underline{n}(\boldsymbol{y}_j), \quad j = 1, \ldots, J \tag{28.7}$$

$$\underline{h}(\boldsymbol{z}_k) = (s\, f(\boldsymbol{x}_k)) + t) + \underline{m}(\boldsymbol{z}_k), \quad k = 1, \ldots, K \tag{28.8}$$

With the geometric transformations (28.2) we thus explicitly have

$$\underline{g}(\boldsymbol{y}_j) = 1/s \cdot \left(f\left(B\boldsymbol{y}_j + \boldsymbol{b}\right) - t\right) + \underline{n}(\boldsymbol{y}_j), \quad j = 1, \ldots, J \tag{28.9}$$

$$\underline{h}(\boldsymbol{z}_k) = \left(s\, f\left(B^{-1}(\boldsymbol{z}_k - \boldsymbol{b})\right) + t\right) + \underline{m}(\boldsymbol{z}_k), \quad k = 1, \ldots, K . \tag{28.10}$$

The reference image f is assumed to be a smooth function. We represent it using a regular grid with bi-cubic interpolation, which is necessary, since the coordinates $\boldsymbol{x}_j = B\boldsymbol{y}_j + \boldsymbol{b}$ and $\boldsymbol{x}_k = B^{-1}(\boldsymbol{z}_k - \boldsymbol{b})$ are real valued and generally do not fall on the grid.

28.2.2 The estimation

The task is to estimate the eight parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_G, \boldsymbol{\theta}_I)$ for the geometric and the radiometric transformation and the unknown true signal f from the observed values $g(\boldsymbol{y}_j)$ and $h(\boldsymbol{z}_k)$.

The explicit modeling in (28.9) and (28.10) allows us to write the problem as nonlinear Gauss-Markov model with the residuals

$$n_j(\boldsymbol{\theta}, f) = g_j - 1/s \cdot \left(f\left(B\boldsymbol{y}_j + \boldsymbol{b}\right) - t\right) \quad , \qquad \mathbb{D}(\underline{n}_j) = \sigma_{n_j}^2 , \quad j = 1, ..., J \tag{28.11}$$

$$m_k(\boldsymbol{\theta}, f) = h_k - \left(s\, f\left(B^{-1}(\boldsymbol{z}_k - \boldsymbol{b})\right) + t\right) , \qquad \mathbb{D}(\underline{m}_k) = \sigma_{m_k}^2 , \quad k = 1, ..., K \tag{28.12}$$

for all pixels $g_j := g(\boldsymbol{y}_j)$ of $\boldsymbol{g}$ and all pixels $h_k := h(\boldsymbol{z}_k)$ of $\boldsymbol{h}$ falling into the common region in $\boldsymbol{f}$. Maximum likelihood (ML) estimates result from minimizing the weighted sum of the residuals

$$\Omega(\boldsymbol{\theta}, f) = \sum_j w_j n_j^2(\boldsymbol{\theta}, f) + \sum_k w_k m_k^2(\boldsymbol{\theta}, f) \tag{28.13}$$

w.r.t. the unknown distortion parameters $\boldsymbol{\theta}$ and the unknown signal f, using proper weights

$$w_j = \frac{1}{\sigma_{n_j}^2} \quad \text{and} \quad w_k = \frac{1}{\sigma_{n_k}^2} \tag{28.14}$$

The statistical properties of the noise need to be specified, e.g., assuming the variance to be signal dependent, thus e.g.,

$$\sigma_n^2(\boldsymbol{y}) = V_g(\tilde{g}(\boldsymbol{y})) \qquad \sigma_m^2(\boldsymbol{z}) = V_h(\tilde{h}(\boldsymbol{z}))\,. \tag{28.15}$$

We prior to the optimization estimate the signal dependent variance functions of the two images, see Förstner (2000). In addition we take the effect of bi-linear interpolation into account, see (28.18).

Since, due to the size of f, the number of unknowns is comparably large, say, in the range between 200 and 10000. Therefore, we solve this problem by alternatively fixing one group of the parameters and solving for the other:

$$\hat{\boldsymbol{\theta}} \mid \hat{f} \;=\; \text{argmin}_\theta \Omega(\boldsymbol{\theta}, \hat{f})\,, \tag{28.16}$$

$$\hat{f} \mid \hat{\boldsymbol{\theta}} \;=\; \text{argmin}_f \Omega(\hat{\boldsymbol{\theta}}, f)\,, \tag{28.17}$$

in a block Gauss-Seidel fashion.

Especially, the estimated unknown function f is the weighted mean of the functions g and h transformed into the coordinate system $\boldsymbol{x}$ of f, which can be calculated pixel wise as a weighted sum of the two image windows warped into f:

$$\hat{f}_i \mid \hat{\boldsymbol{\theta}} = \frac{w_{f_i}^{(g)}\, f_i^{(g)} \;+\; w_{f_i}^{(h)}\, f_i^{(h)}}{w_{f_i}^{(g)} \;+\; w_{f_i}^{(h)}}\,, \tag{28.18}$$

with the warped image windows

$$f_i^{(g)} := f_i^{(g)}(\boldsymbol{x}_i) = s \cdot g(\boldsymbol{y}_i) + t \quad \text{and} \quad f_i^{(h)} := f_i^{(h)}(\boldsymbol{x}_i) = 1/s \cdot (h(\boldsymbol{z}_i) - t) \tag{28.19}$$

from (28.3) and

$$\boldsymbol{y}_i = B^{-1}(\boldsymbol{x}_i - \boldsymbol{b})) \quad \text{and} \quad \boldsymbol{z}_i = B\boldsymbol{x}_i + \boldsymbol{b} \tag{28.20}$$

from (28.2). The weights are

$$w_{f_i}^{(g)} = \frac{1}{s^2 \cdot V_g(g(\boldsymbol{y}_i))} \quad \text{and} \quad w_{f_i}^{(h)} = \frac{s^2}{V_g(h(\boldsymbol{z}_i))}\,. \tag{28.21}$$

Bi-cubic interpolation is used to transfer $g(\boldsymbol{y}_i)$ and $h(\boldsymbol{z}_i)$ to $f(\boldsymbol{x}_i)$, see (28.19). Observe, the individual pixels f_i of the estimated function generally do not lie in the middle of corresponding pixels $f_i^{(g)}$ and $f_i^{(h)}$.

Bi-cubic interpolation induces additional errors, which we interpret as additional noise of the observations. Hence, the variances of the observations are assumed to have two components, one from the imaging process and one, the variances $\sigma_{\delta_g}^2$ and $\sigma_{\delta_h}^2$, from the interpolation process:

$$\sigma_n^2(\boldsymbol{y}) = V_g(\tilde{g}(\boldsymbol{y})) + 1/s \cdot \sigma_{\delta_g}^2 \qquad \sigma_m^2(\boldsymbol{z}) = V_h(\tilde{h}(\boldsymbol{z})) + s\;\sigma_{\delta_h}^2\,. \tag{28.22}$$

For details for the interpolation error see the Appendix, Eq. (28.129).

As a result of the ML-estimation we obtain: (1) the parameters $\hat{\boldsymbol{\theta}}$, (2) their $\Sigma_{\hat{\theta}\hat{\theta}}$, and (3) the variance factor ,

$$\hat{\sigma}_0^2 = \frac{\Omega(\hat{\boldsymbol{\theta}}, \hat{f})}{R}\,, \tag{28.23}$$

where R is the redundancy of the system, i.e., the efficient number of observations[1] $K_g + K_h$ minus the number of unknown parameters $8 + K_f$, where we take the approximation $K_f = \sqrt{K_g K_h}$ for the number of parameters of the grid defining f:

$$R = K_g + K_h - (8 + \sqrt{K_g K_h})\,. \tag{28.24}$$

[1] We set $K_g := J$ and $K_h := K$.

If the model holds, the variance factor is Fisher distributed with $F(R,\infty)$, thus should be close to 1. Therefore, it is reasonable to multiply the covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$ with the variance factor to arrive at a realistic characterization

$$\hat{\Sigma}_{\hat{\theta}\hat{\theta}} = \hat{\sigma}_0^2 \, \Sigma_{\hat{\theta}\hat{\theta}} \tag{28.25}$$

of the uncertainty of the estimated parameters.

The covariance matrix $\hat{\Sigma}_{\hat{\psi}\hat{\psi}}$ of the sought affinities finally are derived by variance propagation from

$$\mathsf{A}(\boldsymbol{\psi}_{1..6}) = \begin{bmatrix} \psi_1 & \psi_3 & \psi_5 \\ \psi_2 & \psi_4 & \psi_6 \\ 0 & 0 & 1 \end{bmatrix} = \mathsf{B}(\boldsymbol{\theta}_{1..6})^2 = \begin{bmatrix} \theta_1^2 + \theta_2\theta_3 & \theta_1\theta_3 + \theta_3\theta_4 & \theta_5 + \theta_1\theta_5 + \theta_3\theta_6 \\ \theta_1\theta_2 + \theta_2\theta_4 & \theta_4^2 + \theta_2\theta_3 & \theta_6 + \theta_2\theta_5 + \theta_4\theta_6 \\ 0 & 0 & 1 \end{bmatrix} \tag{28.26}$$

and

$$\mathsf{R}(\boldsymbol{\psi}_{7,8}) = \begin{bmatrix} \psi_7 & \psi_8 \\ 0 & 1 \end{bmatrix} = \mathsf{S}(\boldsymbol{\theta}_{7,8})^2 = \begin{bmatrix} \theta_7^2 & \theta_8 + \theta_7\theta_8 \\ 0 & 1 \end{bmatrix} \tag{28.27}$$

with the Jacobian

$$J_{\psi\theta} = \begin{bmatrix} 2\,\theta_1 & \theta_3 & \theta_2 & 0 & 0 & 0 & 0 & 0 \\ \theta_2 & \theta_1 + \theta_4 & 0 & \theta_2 & 0 & 0 & 0 & 0 \\ \theta_3 & 0 & \theta_1 + \theta_4 & \theta_3 & 0 & 0 & 0 & 0 \\ 0 & \theta_3 & \theta_2 & 2\,\theta_4 & 0 & 0 & 0 & 0 \\ \theta_5 & 0 & \theta_6 & 0 & \theta_1 + 1 & \theta_3 & 0 & 0 \\ 0 & \theta_5 & 0 & \theta_6 & \theta_2 & \theta_4 + 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\,\theta_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_8 & \theta_7 + 1 \end{bmatrix}, \tag{28.28}$$

hence

$$\hat{\Sigma}_{\hat{\psi}\hat{\psi}} = \hat{\sigma}_0^2 \, J_{\psi\theta} \, \Sigma_{\hat{\theta}\hat{\theta}} \, J_{\psi\theta}^{\mathsf{T}} \, . \tag{28.29}$$

28.2.3 Refined correspondences

The result of the LSM can be used to provide refined correspondences.

28.2.3.1 Point and affine correspondences

Point correspondences follow from (28.26). For some arbitrary point $\boldsymbol{y}_0$ in the left image we obtain its correspondent coordinates in the right image from

$$\boldsymbol{z} = \hat{A} \, \boldsymbol{y}_0 + \hat{\boldsymbol{c}} \tag{28.30}$$

with

$$\hat{A} = \begin{bmatrix} \hat{\psi}_1 & \hat{\psi}_3 \\ \hat{\psi}_2 & \hat{\psi}_4 \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{c}} = \begin{bmatrix} \hat{\psi}_5 \\ \hat{\psi}_6 \end{bmatrix} . \tag{28.31}$$

Thus, we have the corresponding point pair

$$\boxed{\{\boldsymbol{y}_0, \boldsymbol{z}\} \quad \text{with} \quad \boldsymbol{z} = \hat{A} \, \boldsymbol{y}_0 + \hat{\boldsymbol{c}} .} \tag{28.32}$$

If the centre point in the left image is taken, we have $\boldsymbol{y}_{00} = \boldsymbol{0}$ and the corresponding point pair is

$$\boxed{\{\boldsymbol{y}_{00}, \boldsymbol{z}\} \quad \text{with} \quad \boldsymbol{y}_{00} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{z} = \hat{\boldsymbol{c}} .} \tag{28.33}$$

Affine correspondences are defined as a pair $\{\boldsymbol{y}_0, \boldsymbol{z}_0\}$ of corresponding keypoints together with the affine transform $\hat{\mathsf{A}}$ in (28.26) which – as seen above, see (28.30) – can be used to refine the position of $\boldsymbol{z}_0$ as in (28.30) together with the local distortion:

$$\boxed{\left\{\boldsymbol{y}_0, \boldsymbol{z}_0, \hat{\mathsf{A}}\right\} .} \tag{28.34}$$

28.2.3.2 Uncertain point correspondences

The covariance matrix of the estimated parameters can be used to derive a pair of corresponding points together with their uncertainty. In case we choose the centre $\boldsymbol{y}_{00} = \mathbf{0}$ of the left window as point to be matched we have the following uncertain correspondence

$$\boxed{\{\boldsymbol{y}_{00}, \boldsymbol{z}, \Sigma_{zz}\}} \tag{28.35}$$

with

$$\boldsymbol{y}_{00} = \mathbf{0}\,, \quad \boldsymbol{z} = \widehat{\boldsymbol{c}} = \begin{bmatrix} \widehat{\psi}_5 \\ \widehat{\psi}_6 \end{bmatrix} \quad \text{and} \quad \Sigma_{zz} = \Sigma_{\widehat{c}\widehat{c}} = \begin{bmatrix} \sigma^2_{\widehat{\psi}_5} & \sigma_{\widehat{\psi}_5}\sigma_{\widehat{\psi}_6} \\ \sigma_{\widehat{\psi}_6}\sigma_{\widehat{\psi}_5} & \sigma^2_{\widehat{\psi}_6} \end{bmatrix} \tag{28.36}$$

Observe, the point in the left image is assumed to be certain, while the point in the right image carries all uncertainty. The covariance matrix $\Sigma_{\hat{a}\hat{a}}$ therefore also provides the uncertainty of the parallaxes $\boldsymbol{p} = \boldsymbol{z} - \boldsymbol{y}_{00} = \boldsymbol{c}$:

$$\Sigma_{pp} = \Sigma_{\widehat{c}\widehat{c}}\,. \tag{28.37}$$

28.3 Realization of symmetric least squares matching

In this section we derive the Jacobians in detail.

28.3.1 Jacobians of g

We start with the differential for

$$g_j = 1/\theta_7 \cdot (f(\boldsymbol{x}_j) - \theta_8) \quad \text{with} \quad \boldsymbol{x}_j = B\boldsymbol{y}_j + \boldsymbol{b} \tag{28.38}$$

w.r.t. to the unknown parameters $\boldsymbol{\theta}$. We explicitly have

$$g_j = 1/\theta_7 \cdot (f(\theta_1 x_j + \theta_3 y_j + \theta_5, \theta_2 x_j + \theta_4 y_j + \theta_6) - \theta_8)\,. \tag{28.39}$$

Assuming approximate values $\boldsymbol{\theta}^a$ for the parameters, we have

$$g_j = g_j^a + \mathrm{d}g_j = g_j^a + \left.\frac{\partial g_j}{\partial \boldsymbol{\theta}}\right|_{\theta^a} \mathrm{d}\boldsymbol{\theta}\,, \tag{28.40}$$

with

$$g_j^a = 1/\theta_7^a \cdot \left(f(B^a\boldsymbol{y}_j + \boldsymbol{b}^a) - \theta_8^a\right)\,, \tag{28.41}$$

and

$$\frac{\partial g_j}{\partial \boldsymbol{\theta}} = 1/\theta_7 \cdot [f_{x,j}x_j \mid f_{y,j}x_j \mid f_{x,j}y_j \mid f_{y,j}y_j \mid f_{x,j} \mid f_{y,j} \mid -(f_j - \theta_8)/\theta_7 \mid -1]\,, \tag{28.42}$$

evaluated at the approximate values. This is due to $\mathrm{d}(1/x) = -1/x^2\,\mathrm{d}x$, which yields

$$\begin{aligned}
\mathrm{d}g_j \;=\; & +1/\theta_7 \cdot f_{x,j} \cdot x_j \cdot \mathrm{d}\theta_1 && (28.43)\\
& +1/\theta_7 \cdot f_{y,j} \cdot x_j \cdot \mathrm{d}\theta_2 && (28.44)\\
& +1/\theta_7 \cdot f_{x,j} \cdot y_j \cdot \mathrm{d}\theta_3 && (28.45)\\
& +1/\theta_7 \cdot f_{y,j} \cdot y_j \cdot \mathrm{d}\theta_4 && (28.46)\\
& +1/\theta_7 \cdot f_{x,j} \cdot \mathrm{d}\theta_5 && (28.47)\\
& +1/\theta_7 \cdot f_{y,j} \cdot \mathrm{d}\theta_6 && (28.48)\\
& -1/\theta_7^2 \cdot (f_j - \theta_8) \cdot \mathrm{d}\theta_7 && (28.49)\\
& -1/\theta_7 \cdot\, \mathrm{d}\theta_8\,. && (28.50)
\end{aligned}$$

28.3.2 Jacobians of h

For the derivatives of

$$h_k = \theta_7 \underbrace{f(\boldsymbol{x}_k)}_{f_k} + \theta_8 \quad \text{with} \quad \boldsymbol{x}_k = B^{-1}(\boldsymbol{z}_k - \boldsymbol{b}) \tag{28.51}$$

w.r.t. $\boldsymbol{\theta}$ we need some preparation. The Jacobian of the inverse is

$$\mathrm{d}B^{-1} = -B^{-1}(\mathrm{d}B)B^{-1} \quad \text{with} \quad B^{-1} = \frac{1}{D}\begin{bmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{bmatrix} \quad \text{and} \quad D := |B|\,. \tag{28.52}$$

We in addition use the substitution

$$\boldsymbol{x}'_k = B^{-1}(\boldsymbol{z}_k - \boldsymbol{b})\,. \tag{28.53}$$

These are the coordinates of the point in f corresponding to $\boldsymbol{z}_k$. Hence, we have – only referring to the first four parameters

$$\begin{aligned} \mathrm{d}(B^{-1}(\boldsymbol{z}_k - \boldsymbol{b})) &= -B^{-1}(\mathrm{d}B)B^{-1}(\boldsymbol{z}_k - \boldsymbol{b}) && (28.54)\\ &= -B^{-1}(\mathrm{d}B)\boldsymbol{x}'_k && (28.55)\\ &= -(\boldsymbol{x}'^{\mathsf{T}}_k \otimes B^{-1})\mathrm{d}\boldsymbol{\theta}\,. && (28.56) \end{aligned}$$

We finally have

$$\begin{aligned} \frac{\partial f(\boldsymbol{z}'_k)}{\partial \boldsymbol{\theta}} &= \frac{\partial f(\boldsymbol{z}'_k)}{\partial x'_k}\frac{\partial x'_k}{\partial \boldsymbol{\theta}} + \frac{\partial f(\boldsymbol{z}'_k)}{\partial y'_k}\frac{\partial y'_k}{\partial \boldsymbol{\theta}} && (28.57)\\ &= - \quad \nabla^{\mathsf{T}} f\ (\boldsymbol{x}'^{\mathsf{T}}_k \otimes B^{-1}) && (28.58)\\ &= -(1 \otimes \nabla^{\mathsf{T}} f)(\boldsymbol{x}'^{\mathsf{T}}_k \otimes B^{-1}) && (28.59)\\ &= -\boldsymbol{x}'^{\mathsf{T}}_k \otimes \nabla^{\mathsf{T}} f B^{-1} && (28.60) \end{aligned}$$

Therefore with

$$\nabla\alpha = \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = B^{-\mathsf{T}}\nabla f \tag{28.61}$$

we obtain

$$\frac{\partial f(\boldsymbol{z}'_k)}{\partial \boldsymbol{\theta}} = -\boldsymbol{x}'^{\mathsf{T}}_k \otimes \nabla^{\mathsf{T}}\alpha_k = -[x'_k\alpha_{x,k} \quad x'_k\alpha_{y,k} \quad y'_k\alpha_{x,k} \quad y'_k\alpha_{y,k}]\,. \tag{28.62}$$

Finally, the differential of h is

$$\begin{aligned} \mathrm{d}h_k &= -\theta_7\alpha_{x,k}x'_k\,\mathrm{d}\theta_1 && (28.63)\\ &\quad -\theta_7\alpha_{y,k}x'_k\,\mathrm{d}\theta_2 && (28.64)\\ &\quad -\theta_7\alpha_{x,k}y'_k\,\mathrm{d}\theta_3 && (28.65)\\ &\quad -\theta_7\alpha_{y,k}y'_k\,\mathrm{d}\theta_4 && (28.66)\\ &\quad -\theta_7\alpha_{x,k}\,\mathrm{d}\theta_5 && (28.67)\\ &\quad -\theta_7\alpha_{y,k}\,\mathrm{d}\theta_6 && (28.68)\\ &\quad +f_k\,\mathrm{d}\theta_7 && (28.69)\\ &\quad +1\,\mathrm{d}\theta_8 && (28.70) \end{aligned}$$

Hene we have the compact form

$$h_k = h^a_k + \mathrm{d}h_k = h^a_k + \frac{\partial h_k}{\partial \boldsymbol{\theta}}\,\mathrm{d}\boldsymbol{\theta} \tag{28.71}$$

with

$$\frac{\partial h_k}{\partial \boldsymbol{\theta}} = -\theta_7\left[\alpha_{x,k}x_k \mid \alpha_{y,k}x_k \mid \alpha_{x,k}y_k \mid \alpha_{y,k}y_k \mid \alpha_{x,k} \mid \alpha_{y,k} \mid -f_k\theta_7^{-1} \mid -1\right]\,, \tag{28.72}$$

evaluated at the approximate values $\boldsymbol{\theta}^a$ and

$$h^a_k = \theta^a_7 f\left((B^a)^{-1}(\boldsymbol{z}_k - \boldsymbol{b}^a)\right) + \theta^a_8\,. \tag{28.73}$$

28.3.3 Jacobian $\boldsymbol{X}$ and the linearized observations $\Delta \boldsymbol{y}$

With the abbreviations

$$\boldsymbol{x}_{g_j\theta}^\mathsf{T} = \frac{\partial g_j}{\partial \boldsymbol{\theta}} \quad \text{and} \quad \boldsymbol{x}_{h_k\theta}^\mathsf{T} = \frac{\partial h_k}{\partial \boldsymbol{\theta}} \tag{28.74}$$

the design matrix $\mathsf{X} = \partial \boldsymbol{y}/\partial \boldsymbol{\theta}$ is given by

$$\mathsf{X} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \mathsf{X}_g \\ \mathsf{X}_h \end{bmatrix} \quad \text{with} \quad \mathsf{X}_g = \begin{bmatrix} \boldsymbol{x}_{g_1\theta}^\mathsf{T} \\ \dots \\ \boldsymbol{x}_{g_j\theta}^\mathsf{T} \\ \dots \\ \boldsymbol{x}_{g_J\theta}^\mathsf{T} \end{bmatrix} \quad \text{and} \quad \mathsf{X}_h = \begin{bmatrix} \boldsymbol{x}_{h_1\theta}^\mathsf{T} \\ \dots \\ \boldsymbol{x}_{h_k\theta}^\mathsf{T} \\ \dots \\ \boldsymbol{x}_{h_K\theta}^\mathsf{T} \end{bmatrix} \tag{28.75}$$

Similarly, we have the residuals or the negative linearized observation

$$\boldsymbol{v} = -\Delta \boldsymbol{y} = -\begin{bmatrix} \Delta \boldsymbol{y}_g \\ \Delta \boldsymbol{y}_h \end{bmatrix} = \begin{bmatrix} [g_j^a - g_j] \\ [h_k^a - h_k] \end{bmatrix} . \tag{28.76}$$

This leads to the normal equation system

$$\mathsf{N}\widehat{\Delta\boldsymbol{\theta}} = \boldsymbol{n} \quad \text{with} \quad \mathsf{N} = \mathsf{X}^\mathsf{T}\mathsf{W}\mathsf{X} \quad \text{and} \quad \boldsymbol{n} = \mathsf{X}^\mathsf{T}\mathsf{W}\Delta\boldsymbol{y} , \tag{28.77}$$

and the updates in the ν-th iteration

$$\widehat{\boldsymbol{\theta}}^{(\nu+1)} = \widehat{\boldsymbol{\theta}}^{(\nu)} + \widehat{\Delta\boldsymbol{\theta}}^{(\nu)} . \tag{28.78}$$

28.3.4 The accuracy potential of LSM

The potential of the refinement of the affinity using LSM, namely the expected precision of the affinity for ideal cases, can be easily derived, see Barath et al. (2020a). This is based on the part of the normal equation matrix N related to the 6 parameters of the geometric affinity: $\mathsf{N} = \sigma_n^{-2} \sum_{ij} \nabla f_\theta(i,j) \nabla f_\theta^\mathsf{T}(i,j)$, where the sum is over all pixels in an $N \times N$ window. If we assume the distortion is zero and f is known, then the gradient is $\nabla f = [xf_x, yf_x, xf_y, yf_y, f_x, f_y]$, see (28.43) ff. Observe, the 2×2 matrix referring to the translation parameters is proportional to the structure tensor of the patch. We now assume that the gradients in the window have the same variance $\sigma_{f'}^2$ and are mutually uncorrelated. Then the normal equation matrix will be diagonal leading to the covariance matrix

$$\Sigma_{\alpha\alpha} = \begin{bmatrix} \sigma_a^2 \mathsf{I}_4 & \mathsf{0} \\ \mathsf{0} & \sigma_p^2 \mathsf{I}_3 \end{bmatrix} \quad \text{with} \quad \sigma_a = \frac{\sqrt{12}}{N^2} \frac{\sigma_n}{\sigma_{f'}} \quad \text{and} \quad \sigma_p = \frac{1}{N} \frac{\sigma_n}{\sigma_{f'}} . \tag{28.79}$$

Hence, the standard deviations of the estimated affinity $\widehat{A}$ and shift $\widehat{\boldsymbol{c}}$ are below 1% and 0.1 pixels, except for very small scales. Moreover, for the window size $M \times M$, the standard deviations decrease with on average with M^2 and M, respectively, see Barath et al. (2020b)

28.3.5 The algorithm

The algorithm `LSM_62_sym_main.m` is given below.

The input, the output and the algorithmic steps are the following:

- The two image windows must be square and have values in the range $[0, 255]$. They have sizes $M_g \times M_g$ and $M_h \times M_h$, where the widths M_g and M_h are odd numbers. Their centres are at $[N_g + 1, N_g + 1]$ and $[N_h + 1, N_h + 1]$ with

$$N_g = \frac{M_g - 1}{2} \quad \text{and} \quad N_h = \frac{M_h - 1}{2} . \tag{28.80}$$

  The centres of the image windows – with coordinates $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ in the image – are assumed to be given by some keypoint detector, possibly with rounded coordinates.

Algorithm 5: Symmetric Least Squares Matching.
$[\widehat{\boldsymbol{\theta}}, \Sigma_{\hat{\theta}\hat{\theta}}, \hat{\sigma}_0^2, R] = \{\texttt{Sym\_LSM}\}(g, h, V_g, V_h, \mathsf{A}^a, \mathsf{R}^a, \sigma_s, \max_\nu)$
Input: observed image patches $\{g, h\}$, must be square with odd width
variance functions V_g, V_h
approximate transformations $\mathsf{A}^a, \mathsf{R}^a$
smoothing parameter σ_s
maximum number of iterations $\max_\nu$.
Output: estimated parameters $\{\widehat{\boldsymbol{\theta}}, \Sigma_{\hat{\theta}\hat{\theta}}\}$;
variance factor $\hat{\sigma}_0^2$;
redundancy R.

```
1  Initial approximate transformations/parameters: B^a = (A^a)^{1/2}, S^a = (R^a)^{1/2};
2  set iterations ν = 0, approximate parameters θ̂^0 := θ̂^a;
3  repeat
4  |  iteration ν := ν + 1;
5  |  find overlap N_f and observing pixels 𝒴, 𝒵 with weights w;
6  |  determine estimate f̂ of true signal, possibly smoothed with G(σ_s);
7  |  determine derivatives f̂_x and f̂_y;
8  |  foreach i = {g, h} do
9  |  |  warp [f̂, f̂_x, f̂_y] into i;
10 |  |  forall p ∈ i do
11 |  |  |  determine f̂(p), f̂_x(p), f̂_x(p);
12 |  |  |  determine Δy and x_iθ;
13 |  |  end
14 |  end
15 |  build system NΔθ = n with N = X^T W X, n = X^T W Δy;
16 |  determine estimates Δ̂θ and new approximate values θ̂^{ν+1};
17 |  determine covariance matrix Σ_θ̂θ̂ = N^{-1};
18 |  determine the redundancy R and the variance factor σ̂_0^2;
19 until max_u(||Δ̂θ_u||/σ_θ̂u) < 0.1 or ν = max_ν;
20 derive transformations Â and R̂ and parameters ψ̂ with Σ_ψ̂ψ̂
```

The affinity refers to a coordinate systems S_y and S_z parallel to the image coordinate system located at these centres. The coordinate systems are right handed, with x =rows and y =columns – in contrast to the MATLAB-image convention.

- The geometric affinity must not change the sign or the chirality of the coordinate system, i.e., must not contain a mirroring or an exchange of the two axes. This is checked internally. In case the affinity between the two images is not sign preserving, one of the images needs to be mirrored before calling the matching routine.
- The variance functions $V_g = V_g(g)$ and $V_h = V_h(h)$ provide the variances as a function of the intensities. These functions may be derived by some, blindÂ´ noise estimation procedure, e.g. `noise_standard_deviation_estimation.m`, which yields the noise standard deviation as a function of the intensity without requiring ground truth. Interpolation errors can be taken into account in this function.
- The approximate affinities for the geometric and the radiometric transformation are to be given as 3×3 matrix and 1×2 vector.

line 1 Internally half of the approximate transformations is used.

line 2 The eight parameters in $\boldsymbol{\theta}$ refer to the six geometric parameters column wise and the two radiometric parameters.

line 5 `LSM_62_sym_par_find_observation_positions.m`: The overlap of the two images

yields a square image f with size $M_f \times M_f$ (odd) with centre at $[N_f + 1, N_f + 1]$ being the origin of the coordinate system S_x, where the geometric coordinate transformations refer to. Again, the half size is $N_f = (M_f - 1)/2$. Only those pixels in g and h which fall into the common overlap after transformation into the coordinate system of f are used. Their coordinates are stored in $\mathcal{Y}$(`yi`) and $\mathcal{Z}$(`zi`). In order to obtain stable results and avoid oscillations, the overlap is not changed if the parameters change less than 50% of their standard deviations.

line 6 `LSM_62_sym_par_estimate_f.m`: The estimated signal $\widehat{f}$ in a first instance is the weighted average of g and h, transformed into the coordinate system S_x. It may be smoothed with a Gaussian having smoothing kernel σ_s.

7–18 `LSM_62_sym_par_estimate_parameters.m`.

lines 7 The derivatives $\widehat{f}_x$ and $\widehat{f}_y$ are realized by convolutions with Scharr's improved Sobel operator, see Jähne (1999, Vol. II, p. 223)

$$d_x = \frac{1}{32} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} [3 \;\; 10 \;\; 3] \quad \text{and} \quad d_y = d_x^{\mathsf{T}} \,. \tag{28.81}$$

(see also https://en.wikipedia.org/wiki/Sobel_operator, Alternative operators).

line 9 The three images $[\widehat{f}, \widehat{f}_x, \widehat{f}_y]$ are simultaneously warped into the images g and h to directly access the values in the coordinate systems S_y and S_z.

line 11 All function values and their derivatives are interpolated at the observing pixels, stored in $\mathcal{Y}$ and $\mathcal{Z}$.

line 12 The linearized observation $\widehat{\Delta y}$ refers to the original pixels in g and h and the interpolated pixels in f. This allows to determine the Jacobian $\boldsymbol{X}$ for all pixels stored in $\mathcal{Y}$ and $\mathcal{Z}$.

line 16 The parameters are corrected additively.

line 18 The redundancy needs an explanation. We have $K_g = |\mathcal{Y}|$ and $K_h = |\mathcal{Z}$ observations in g and h. The unknowns are the eight parameters $\boldsymbol{\theta}$ and the unknown function f. We assume, that the function f can be represented by the geometric mean of K_g and K_h parameters, which reflects the fact that if the affinity is a pure scaling, say by s, the numbers K_g and K_h differ by s^2. Therefore, we determine the redundancy by

$$R = (K_g + K_h) - (8 + \sqrt{K_g K_h}) \,. \tag{28.82}$$

line 20 The full transformations are determined by squaring the estimated transformations together with their covariance matrix.

28.3.6 Realizing image warpings

Two steps in the algorithm require to warp images:

1. the warping of g and h into f in order to determine the estimate $\widehat{f}$.
2. the warping of f and its derivatives for building the design matrix $\boldsymbol{X}$ and the lienarized observations $\Delta\boldsymbol{y}$.

We provide two realizations. They differ in speed:

1. The first realization[2] performs the warping in a loop over all pixels. Due to the interpreter-characteristics of MATLAB this is slow.

[2] in `LSM_62_sym_estimate_f` and `LSM_62_sym_estimate_parameters`

2. The second realization[3] performs the warping using the function `imtransform.m`, see the blue and green boxes in Fig. 1. The MATLAB function `imtransform.m` is optimized which leads to lower CPU-times.

A transfer to a language which allows to compile the code, like `C`, preferably uses the first realization, since only the necessary pixels are handled. Furthermore, the set of pixels used for matching could be restricted to those, where the gradient is above a, possibly local, threshold in order to further speed up the algorithm.

28.4 Checking the implementation

We can check the coherence of the assumed model and the implementation using simulated data. For this purpose, we start from some ideal, true observation, i.e., image windows, and add noise according to the assumed model to all pixels. Varying the noise, we obtain K samples for the observations, and consequently $k = 1, ..., K$ samples for the estimates.[4]

We then can perform three checks, which can be realized as statistical tests:

1. The K estimated variance factors $\widehat{\sigma}_{0,k}^2$ should on an average be 1, since if the observed images are noisy affinely distorted versions of the true image, the expectation of the variance factors is 1. Deviations of the mean variance factor from 1 indicates, that the assumed model may not hold. If the variance factors do not differ from 1 too much, there is no reason to doubt the underlying model. The mean estimated variance factor follows a F-distributed test statistic F.
2. The K samples lead to K estimates $\boldsymbol{\theta}_k$ of the unknown parameters. The variation of these parameters, captured in their empirical covariance matrix should be close to the theoretical covariance matrix derived from estimation process, which is determined by variance propagation. The difference between the empirical and the theoretical covariance matrix is measured by a χ^2-distributed test statistic X. If this test statistic is not in the rejection region, we have no reason not to use the theoretical covariance matrix as substitute for the empirical covariance matrix.
3. The K estimates $\boldsymbol{\theta}_k$ of the unknown parameters should on an average be close to the given/true parameters specified by the simulation. The bias, i.e., the difference between the mean of the parameters and the true values, leads to a χ^2-distributed test statistic X. If this test statistic is not in the rejection region, we have no reason to assume the estimates are biased.

The interpretation of the three test statistics assumes the model underlying the estimation procedure does not contain any approximations and the software implementation of the estimation model is perfect. Hence, if the tests to not fire, i.e., the test statistics are not in the rejection reason, we can assume both, possible approximations in the model are small and the implementation does not contain (severe) errors. This is the value of these tests.

If, however, the test statistics are in the rejection region, this indicates either the model is valid only approximately or the implementation contains errors.

In our case, there are several small approximations in the estimation process, e.g., the bi-cubic interpolation only leads to approximated interpolated values, since the true underlying signal is not known. This refers to the estimated signal f as well to the first derivatives, which need to be determined at non-integer coordinates.

The three tests are quite sensitive, i.e., increasing the number of samples K increases the probability that the test statistics lie in the rejection region.

The theory for these tests is documented in Förstner and Wrobel (2016, Sect. 4.6.8.1), see App. 28.8.3.

[3] in `LSM_62_sym_warp_estimate_f` and `LSM_62_sym_warp_estimate_parameters`

[4] In this context the number K is not to be confused with the number of pixels in the image h.

28.5 Noise variance estimation

The strength of the estimation procedure is to provide the covariance matrix of the estimated parameters which indicates the uncertainty of the estimated geometric and radiometric transformations. In order to this covariance matrix to be realistic, we need realistic variances for the given observations, i.e., the pixels in the two image windows.

In a first approximation we assume the pixels to be statistically independent and the variance of each pixels is a function of the intensity. This is motivated by the Poisson statistics of the photon counts which electronically are transferred into intensities. Since digital cameras generally aim at yielding nicely looking images, the photon counts of the sensors are further processed. This process usually is not made public by the producer of cameras.

Therefore, we assume the variance of the intensities is some arbitrary smooth function of the intensity, and estimate this functions. This is done for each channel of the left and the right image. Since the matching algorithms assumes grey-level images, we take the average variance of the three channels as sufficiently good approximation for the variance function.

The noise variance estimation assumes, that the image contains sufficiently many pixels where the gradient is small, and estimates the noise variance from the gradients in these regions. Therefore, noise estimation should be based on large enough windows, possibly larger than those used for matching.

Hence noise variance estimation can be done in two modes:

1. The noise variance functions $\sigma_n^2(g)$ and $\sigma_n^2(h)$ (variables `vg` and `vh`) are determined from one or several images which are characteristic for the matching windows, and used during the matching process.
2. The noise variance function $\sigma_n^2(g)$ and $\sigma_n^2(h)$ are determined from a large enough region, e.g., 400×400 pixels around the matching window, e.g., if the matching windows are specified by a keypoint in a larger image.

The second alternative is realized in the demos `demo_medium.m` and `demo_image_pairs.m`.

The theory for the noise estimation is given in Förstner (2000, Sect. 3), see App. 28.8.2.

28.6 Demo routines

We have realized four demo-routines,

1. `demo_LSM_small.m` for showing the use of the main routine when two image windows, approximate values and the variacne functions are given,
2. `demo_LSM_medium.m` for showing the use of the main routine when two image windows together with two corresponding Lowe-keypoints are given,
3. `demo_LSM_simulated.m` using simulated data for checking the correctness of the implementation, and
4. `demo_LSM_image_pairs.m` using real data with interactively identified correspondences.

Generally, the control parameters are set in separate files and stored in the struct `par`. The two files are `simulated_set_parameters.m` and `image_pair_set_parameters.m`.

28.6.1 Demo `demo_LSM_small.m`

The demo routine `demo_LSM_small.m` shows the most simple form of using of the main routine, here `LSM_62_sym_warp_main.m`. The required input data are loaded from file. The routine assumes the approximate values for the geometric transformation is given.

Furthermore, the noise variance functions vg and vh are assumed to be provided, e.g. determined from a representative image using the routine `noise_standard_deviation_estimation.m`. The input images, the noise standard deviations and the change of the estimated image windows are shown in figures. It is best to start with this demo.

28.6.2 Demo `demo_LSM_medium.m`

The demo routine `demo_LSM_medium.m` shows how the noise variance estimation is integrated into the matching process. Again, the required input data are loaded from file. Here these are the two complete images together with two corresponding LOWE-keypoints (coordinates, scale, direction). These are used to define the window size and the approximate values. The noise variance functions `vg` and `vh` are determined automatically from the area around the keypoints. First, the input images with the keypoints are shown. When zooming into the keypoints the centre and the direction vector fixing the scale and the direction can be seen. The start and the end of this arrow can be provided interactively, when setting the parameters `par.readX`=1 in the routine `image_pair_set_parameters.m`, line 34. Further figures show the selected windows, the noise standard deviations and the change of the estimated image windows.

28.6.3 Demo `demo_LSM_simulated.m`

The demo routine `demo_LSM_simulated.m` is meant to check the correctness of the implementation based on simulated data. It allows to monitor the individual iterations for a single case, or to statistically test, whether the resultant parameters and their covariance matrix are coherent with the theoretical values, the true values of the parameters and the theoretical covariance matrix derived by the estimation procedure (Cramer-Rao bound).

28.6.3.1 Control parameters

We have the following options, which can be set in the main routine:

- Choosing whether the random sequence is pre-specified or randomly initiated:
  variable `init_rand`
- Choosing the number of samples for checking the covariance matrix and for bias:
  variable `N_samples`
- Choosing between three artificially generated images and taking a window of a given image as reference:
  variable `type_data`

The true transformations and the true images are generated. Within a loop, the true images are perturbed by Gaussian noise and rounded to integers. The `N_samples` are used to test for the correctness of the estimation. In addition we have the following options, which can be set in the routine `simulated_set_parameters.m`:

- Choosing the approximate window size of the overlapping image:
  variable `Nh`
- Choosing whether a test on swapping the two images is to be performed:
  variable `test_symmetry`
- Choosing the geometric and radiometric transformation:
  variables `A_true` and `R_true`
- Choosing the smoothing kernel σ_s:
  variable `sigma_smooth`
- Choosing the maximum number of iterations:
  variable `max_iter`
- Choosing the significance number S of the statistical tests:
  variable `S`

28.6.3.2 Output

The output is different, when looking into the individual iterations (`N_samples` = 1) or when checking the implementation (`N_samples` > 9).

The individual iterations. When analysing the individual iterations for a single sample (`N_samples` = 1), the command window shows the following information

- Document of the control parameters
- Per iteration the number of observations, `N` and the estimated $\hat{\sigma}_0$ as `sigma_0_est`. If $\hat{\sigma}_0$ is not close to 1, this indicates, the assumed model does not fit to the observations. The cause of this effect cannot be given: it may be, that the scene is not flat, the windows are too large, the estimated noise variance deviates from the noise variance in the windows, the shadow situation in both images is different, and so on.
- The final result is characterized by the estimated transformations `A_est` and `R_est`.
- A warning is given, if the maximum number of iterations is reached.
- If the symmetry of the solution is tested, the checks $\hat{\mathsf{A}} \cdot \widehat{\mathsf{A}^{-1}} - I_3$ and $\hat{\mathsf{R}} \cdot \widehat{\mathsf{R}^{-1}} - I_2$ are provided as `check_symmetry_AAi_I` and `check_symmetry_RRi_I`

In addition, the following figures are provided

- the true image $f(\boldsymbol{x})$ (large black box in Fig. 28.2)
- the true mean, left, and right images $f(\boldsymbol{x})$, $g(\boldsymbol{y})$ and $h(\boldsymbol{z})$, respectively (the small black box, the blue box, and the green box in Fig. 28.2)
- the noisy image window pair
- for each iteration, left and the right image warped into the x-coordinate system, i.e., $f(\boldsymbol{y})$ and $f(\boldsymbol{z})$ (the part of the blue and the green parallelograms lying within the small black box in Fig. 28.2)[5].

Checking the implementation. When checking the implementation, thus N_samples > 9, the command window shows the following information

- Document of the control parameters
- Monitoring the samples
- The number of cases, where the maximum number of iterations is reached is documented as a warning. These are not used for the following analysis.
- The result of the statistical tests for all 8 parameters and only the 6 geometric parameters:
  1. test whether the mean of the estimated variance factor deviates from 1,
  2. test whether the empirical covariance matrix $\hat{\Sigma}_{\hat{\theta}\hat{\theta}}$ of the parameters, derived from the estimates $\hat{\boldsymbol{\theta}}$, coincides with the theoretical covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$ from the inverse normal equation matrix,
  3. test whether the mean of the estimated parameters is identical to the true (simulated) value,

  see Förstner and Wrobel (2016, Chapt. 4.6.8). Test statistics lying in the rejection region are indicated with `*****`. Actually, the non-rejection region is given.
- For each parameter, the theoretical and the empirical standard deviation, their ratio, the mean, the standard deviation, and the maximum bias.
- The average standard deviation of the parameters of the affinity, the translation and the radiometric transformation.

[5] The size of the image patches depends on the maximum number of iterations.

- Information about the CPU time.

In addition there are figures showing a noisy sample image window pair, the histograms of the estimated variance factors (twice) and the number of iterations.

28.6.4 Demo `demo_LSM_image_pairs.m`

The routine is meant to apply the matching routine to real data. The user may choose to interactively measure the correspondences or read the previously measured data from file in folder `Images`. For each image pair the correspondences are stored in a `mat`-file in the folder `Data/ImageCoordinates`. Color images are converted to black and white images.

The user is asked to identify two corresponding points together with a scale σ and orientation, mimicking the output of the Lowe-detector. The sequence of actions is the following. For each image

- identify an approximate position in the image,
- a blow-up of the surrounding point is provided,
- the first point to be measured is the centre of the window,
- the second point to be measured provides *three times* the dominant scale σ and the direction.

Windows of size $8\sigma \times 8\sigma$ around the measured keypoints are used for matching. The signal dependent noise variances for both images are estimated from a larger neighbourhood ($\leq 200 \times 200$ pixels) and used for defining the weights of the intensities.

28.6.5 Error messages and convergence

The following error messages may occur:

- `Geometric affinity is not positive definite`. The approximate affinity must be represented by a positive definite 2×2 matrix A. No mirroring is allowed.
- `Overlap is too small`. The overlap of the two images in each iteration must lead to a square window of at least 9×9.

In both cases the output parameter `Red` of the main-routine is negative.

Convergence is guaranteed if the number `N_iter` of used iterations is smaller than the maximum number `max_iter` of iterations. If the number of used iterations is identical to the maximum number of iterations, and the maximum relative change `max_ratio`$= \max_u(||\widehat{\Delta\theta}_u^{(\nu)}||/\sigma_{\hat{\theta}})$ of the parameters in the last iteration is smaller than 1, then convergence can be assumed. Generally, there is no guarantee, that the global optimum is reached.

28.7 Timing

The time mainly depends on the size of the images, i.e., the average number N of pixels in the two images. On an Lenovo X220 with Matlab 2018 we have found the following approximate relation between the number of pixels and the time per iteration, depending on whether the warping function of MATLAB is used or the design matrix is built up using loops on the individual pixels:

$$t_{\text{warping}}\ [\text{ms}] = (0.0047\ N + 12)\ [\text{ms}] \quad \text{and} \quad t_{\text{loop}}\ [\text{ms}] = (0.050\ N + 6.2)\ [\text{ms}]\,. \quad (28.83)$$

Hence, when using the warping function of MATLAB, the computing time takes below 0.05 milliseconds/pixel. With usually 3 iterations, windows of 40×40 can be matched in less than 0.1 seconds.

28.8 Appendix

28.8.1 Bi-cubic interpolation

Bi-cubic interpolation is required for estimating the true underlying function f from the observed images g and h. This interpolation induces errors, which we take into account when specifying the variances of the observed images. Therefore, we analyse the effect of this interpolation onto a signal and derive the relations between the original and the interpolated signal as a basis for variance propagation.

28.8.1.1 1D cubic interpolation

Compact representation. Here we follow Shu (2013). For each value x in the interval $[i, i+1]$ the function is a cubic polynomial which satisfies the following conditions

1. The function at i has the values $f(i) = b_i$
2. The function at $i+1$ has the value $f(i+1) = b_{i+1}$.
3. The derivative at i is $f'(i) = (b_{i+1} - b_{i-1})/2$.
4. The derivative at $i+1$ is $f'(i) = (b_{i+2} - b_i)/2$.

Hence, we need the four neighbouring values collected in

$$\boldsymbol{b} = \begin{bmatrix} b_{i-1} \\ b_i \\ b_{i+1} \\ b_{i+2} \end{bmatrix} . \tag{28.84}$$

We use the substitution

$$\boldsymbol{u}(x) = \begin{bmatrix} 1 \\ (x-i) \\ (x-i)^2 \\ (x-i)^3 \end{bmatrix} \quad \text{with} \quad i = \lfloor x \rfloor \, . \tag{28.85}$$

Then with

$$M_0 = \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} . \tag{28.86}$$

the interpolated value is

$$f(x) = \boldsymbol{u}^\mathsf{T}(x) M_0 \boldsymbol{b} \, , \tag{28.87}$$

as above.

Proof: We use

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} . \tag{28.88}$$

Then cubic function in the i-th interval $[i, i+1]$ can be written as

$$f^{(i)}(u) = a_0 + a_1(u-i) + a_2(u-i)^2 + a_3(u-i)^3 = \boldsymbol{u}^\mathsf{T} \boldsymbol{a} \tag{28.89}$$

The derivative is

$$f_u^{(i)}(u) = a_1 + 2a_2(u-i) + 3a_3(u-i)^2 = \boldsymbol{u} \begin{bmatrix} a_1 \\ 2a_2 \\ 3a_3^2 \\ 0 \end{bmatrix} = \boldsymbol{u}^\mathsf{T} D \boldsymbol{a} \quad \text{with} \quad D = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{28.90}$$

The four conditions then can be written as

$$\begin{bmatrix} f^{(i)}(i) \\ f^{(i)}(i+1) \\ f_u^{(i)}(i) \\ f_u^{(i)}(i+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & -1/2 & 0 & 1/2 \end{bmatrix}}_{U} \begin{bmatrix} b_{i-1} \\ b_i \\ b_{i+1} \\ b_{i+2} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix}}_{V} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{28.91}$$

or compactly

$$\boldsymbol{c} = U\boldsymbol{b} = V\boldsymbol{a}\,. \tag{28.92}$$

Therefore

$$\boldsymbol{a} = V^{-1}U = M_0\boldsymbol{b}\,, \tag{28.93}$$

which holds since

$$U = VM_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & -1/2 & 0 & 1/2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \tag{28.94}$$

◇

Remark: This definition of interpolating cubic splines does not minimize the total curvature of the interpolating function, thus differs from the classical definition. In contrast to the classical definition, the function values $f(u)$ only depend on four neighbouring points b_i linearly, not on all values of the profile. ◇

Since, with $\boldsymbol{a} = M_0\boldsymbol{b}$ the first derivative of the polynomial from (28.90) we obtain the compact expression for the derivative

$$\boxed{f'(x) = \boldsymbol{u}^{\mathsf{T}}(x)M_1\boldsymbol{b}} \tag{28.95}$$

with

$$M_1 = DM_0 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 & 0 \\ 4 & -10 & 8 & -2 \\ -3 & 9 & -9 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{28.96}$$

Variance of cubic interpolation. We now give the variances of cubic interpolation. This is the uncertainty of the interpolated values assuming the given values are uncertain and bi-cubic interpolation is the correct model.

Cubic interpolation at $r \in [0,1]$ requires the vales of g at $[-1,0,1,2]$. Specifically we obtain the interpolated value

$$\begin{aligned} g(r) \quad = \quad & \frac{1}{2}\left[\left(-r + 2\,r^2 - r^3\right) g_{-1}\right. && (28.97)\\ & + \left(2 - 5\,r^2 + 3\,r^3\right) g_0 && (28.98)\\ & + \left(r + 4\,r^2 - 3\,r^3\right) g_1 && (28.99)\\ & \left. + \left(-r^2 + r^3\right) g_2\right]\,, && (28.100) \end{aligned}$$

see (28.86). Assuming homogeneous noise variance, we obtain the variance

$$\sigma_{\overline{n}}^2(r) \quad = \quad q_{\text{cubic}}^2(r)\sigma_n^2 \tag{28.101}$$

with

$$\text{with} \quad q_{\text{cubic}}^2(r) = \frac{1}{2}\left[2 - 9\,r^2 + 8\,r^3 + 21\,r^4 - 30\,r^5 + 10\,r^6\right]\,. \tag{28.102}$$

It is symmetric w.r.t. $r = 1/2$. It reaches its its maximum $\sigma_x^2(0) = \sigma_n^2$ at $r = 0$ and $r = 1$ and its minimum at $r = 1/2$ min

$$\sigma_{\overline{n}}^2(r = 1/2) = \frac{41}{64}\sigma_n^2 \approx 0.641\sigma_n^2\,. \tag{28.103}$$

We may derive an individual variance as a function of the remainder $r = x - \lfloor x \rfloor$. Furtheron, we also can use the average variance, which is

$$\overline{\sigma_{\overline{n}}^2} = \int_{r=0}^{1} \sigma_x^2 \mathrm{d}r = \frac{57}{70}\sigma_n^2 \approx 0.814\sigma_n^2 \,. \tag{28.104}$$

Using this mean value, we have an error of approximately 13% in the variance.

28.8.1.2 2D cubic interpolation

We thus obtain *bi-cubic interpolation* using the substitution

$$\boldsymbol{v}(y) = \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \quad \text{with} \quad v = y - \lfloor y \rfloor \tag{28.105}$$

and the collection of the 4×4 neighbouring values in the cell $[i, i+1] \times [j, j+1]$

$$B = \begin{bmatrix} b_{i-1,j-1} & b_{i-1,j} & b_{i-1,j+1} & b_{i-1,j+2} \\ b_{i,j-1} & b_{i,j} & b_{i,j+1} & b_{i,j+2} \\ b_{i+1,j-1} & b_{i+1,j} & b_{i+1,j+1} & b_{i+1,j+2} \\ b_{i+2,j-1} & b_{i+2,j} & b_{i+2,j+1} & b_{i+2,j+2} \end{bmatrix} , \tag{28.106}$$

We obtain

$$f(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 B M_0^{\mathsf{T}} \boldsymbol{v}(y) \,. \tag{28.107}$$

The partial derivatives then are

$$\boxed{f_x(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_1 B M_0^{\mathsf{T}} \boldsymbol{v}(y) \quad \text{and} \quad f_y(x, y) = \boldsymbol{u}^{\mathsf{T}}(x) M_0 B M_1^{\mathsf{T}} \boldsymbol{v}(y)} \tag{28.108}$$

The variance of bi-cubic interpolated values is

$$\sigma_{\overline{n}}(x, y) = q^2_{\text{bi-cubic}}(r, s)\sigma_n^2 \quad \text{with} \quad q^2_{\text{bi-cubic}}(r, s) = q^2_{\text{cubic}}(r) q^2_{\text{cubic}}(s) \,. \tag{28.109}$$

The average variance is

$$\overline{\sigma_{\overline{n}}^2} = \int_{r=0}^{1} \int_{s=0}^{1} \sigma_{\overline{n}}^2(x, y) \mathrm{d}x \mathrm{d}y = \frac{57^2}{70}\sigma_n^2 \approx 0.663\sigma_n^2 \,. \tag{28.110}$$

Interpolation error. We want to determine the interpolation error of a function $f(x, y)$. Here we assume the data are fixed, i.e., not contaminated by random errors, and the interpolation leads to erroneous results, since the interpolation rule may be different. Since the true interpolation rule is unknown, we perform two bi-cubic interpolations, and compare the result with the original function.

We do this in three steps, see Fig. 28.3:

1. Interpolating the function f at the grid at $[i + 1/2, j + 1/2]$:

$$g(x, y) = f_B(x + 1/2, y + 1/2) \,. \tag{28.111}$$

2. Interpolating the function g at the grid at $[i - 1/2, j - 1/2]$:

$$h(x, y) = g_B(x - 1/2, y - 1/2) \,. \tag{28.112}$$

3. Determining the error induced by the two interpolations

$$\sigma^2 = \mathbb{D}(h(x, y) - f(x, y)) \,. \tag{28.113}$$

We start from (28.107) using

$$u = u(x) - i \quad \text{and} \quad v = v(x) - j\,, \tag{28.114}$$

with the special choice for $x = +1/2$

$$u_+ = x - \lfloor x \rfloor = +1/2 - 0 = 1/2 \quad \text{and} \quad v_+ = 1/2\,. \tag{28.115}$$

Hence we have

$$\boldsymbol{u} = \begin{bmatrix} 1 \\ 1/2 \\ 1/4 \\ 1/8 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v} = \begin{bmatrix} 1 \\ 1/2 \\ 1/4 \\ 1/8 \end{bmatrix} \tag{28.116}$$

We refer to the 49 values of $F(1:7,1:7)$, see Fig. 28.3 This allows to derive $G(1:4,1:4)$

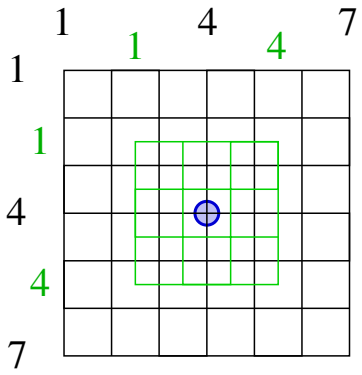


Figure 28.3: Interpolation error. The black 7×7 grid of f is used to interpolate the green 4×4 grid of g. This is used to obtain the interpolated value h at the position of $f(4,4)$. The difference is an indication for double the interpolation error.

via

$$g(i,j) = \boldsymbol{u}^\mathsf{T} M_0 F(i:i+3, j:j+3) M_0^\mathsf{T} \boldsymbol{v}\,. \tag{28.117}$$

Similarly, starting from the interpolated signal g, we can derive the back shifted value

$$h = \boldsymbol{u}^\mathsf{T} M_0 G M_0^\mathsf{T} \boldsymbol{v}\,. \tag{28.118}$$

The difference

$$\boxed{\delta_2(F) = h(F) - f(4,4)} \tag{28.119}$$

is a function of the 49 values of F and represents twice the interpolation error.

We now determine the standard deviation of the interpolation. First, the Jacobian $J_2 = \partial\delta_2/\partial F$ is independent of f and given by

$$J_2 = \frac{1}{2^{16}} \begin{bmatrix} 1 & -18 & 63 & 164 & 63 & -18 & 1 \\ -18 & 324 & -1134 & -2952 & -1134 & 324 & -18 \\ 63 & -1134 & 3969 & 10332 & 3969 & -1134 & 63 \\ 164 & -2952 & 10332 & -38640 & 10332 & -2952 & 164 \\ 63 & -1134 & 3969 & 10332 & 3969 & -1134 & 63 \\ -18 & 324 & -1134 & -2952 & -1134 & 324 & -18 \\ 1 & -18 & 63 & 164 & 63 & -18 & 1 \end{bmatrix} \tag{28.120}$$

We now assume that the signal values f are correlated, depending on the underlying power-spectrum. We assume three different cases

1. Gaussian power spectrum. Then the covariance function is

$$C(d) = \exp(-(d/d_0)^2/2)\,. \tag{28.121}$$

2. Laplacian power spectrum. Then the covariance function is

$$C(d) = \frac{1}{1 + (d/d_0)^2}\,. \tag{28.122}$$

3. The powers spectrum follows the power law, e.g., in the form $P(u) \propto f^{-2}$, or $P(u) = 1/(1+u^2)/\pi$. Then the covariance function has the form

$$C(d) = \exp(-d/d_0)\,. \tag{28.123}$$

The value d_0 controls the smoothness of the signal, larger d_0 leads to smoother functions. Via variance propagation we obtain $\mathbb{V}(\delta_2(f))$. Since this difference results from two interpolations we report

$$\boxed{\sigma_\delta = \sqrt{\frac{\mathbb{V}(\delta_2)}{2}}} \tag{28.124}$$

in the table. We observe:

d_0	P(Gauss)	P(Lapl)	P(Power)
1.0000	0.1855	0.2888	0.3499
1.5000	0.0643	0.1860	0.2947
2.0000	0.0249	0.1180	0.2583
2.5000	0.0112	0.0754	0.2323
3.0000	0.0056	0.0491	0.2128
3.5000	0.0031	0.0327	0.1974
4.0000	0.0019	0.0222	0.1849

Table 28.1: Standard deviation σ_δ of bi-cubic interpolation error for different types of correlations and correlation widths d_0 error

1. the larger d_0, i.e., the smoother the signal, the smaller the interpolation error is.
2. As to be expected, signals with Laplacian power spectrum are rougher than those with Gaussian, and smoother than those with the power spectrum following the power law.

These results refer to the shift $[1/2, 1/2]$ and depend on the assumed stochastical model for the signal. For the use in LSM, we do not want a dependency on the stochastical model for the unknown image function. Since $\boldsymbol{J}_2 = \partial\delta_2/\partial\boldsymbol{F}$, we can use the matrix $\boldsymbol{J}_2$ as filter for deriving δ_2 from f:

$$\boxed{\delta_2 = \boldsymbol{J}_2 * f\,.} \tag{28.125}$$

Obviously, the matrix $\boldsymbol{J}_2$ represents a highpass filter. Hence, we can determine the individual interpolation errors due to forward and backward shifting with interpolation from an arbitrary signal. This allows us to derive the variance of the maximal interpolation error from

$$\boxed{\widehat{\sigma}^2_{\delta,\max} = \tfrac{1}{|\mathcal{R}|}\textstyle\sum_{r\in\mathcal{R}} \delta_r^2\,.} \tag{28.126}$$

Since we are interested in the average interpolation error within the region of one pixel, we determine the mean of the expected variance $\overline{\sigma^2_\delta}$ for all forward and backward shifts $[x, y], x, y \in [0, 1]$. Hence, we can determine the interpolation variance for a specific window f from

$$\widehat{\overline{\sigma^2_\delta}} = k\,\widehat{\sigma}^2_{\delta,\max} \quad \text{with} \quad k = \frac{\overline{\sigma^2_\delta}}{\sigma^2_{\delta,\max}}\,. \tag{28.127}$$

As an example we have the variances of the interpolation error for $d_0 = 1$ and Gaussian covariance function at the grid points $[i, j]/8$, $i, j \in \{0, ..., 8\}$ of the unit interval:

$$[\sigma_{ij}^2] = \frac{1}{1000} \begin{bmatrix} 0 & 9 & 93 & 247 & 327 & 247 & 93 & 9 & 0 \\ 9 & 22 & 111 & 267 & 349 & 267 & 111 & 22 & 9 \\ 93 & 111 & 208 & 368 & 449 & 368 & 208 & 111 & 93 \\ 247 & 267 & 368 & 527 & 608 & 527 & 368 & 267 & 247 \\ 327 & 349 & 449 & 608 & 688 & 608 & 449 & 349 & 327 \\ 247 & 267 & 368 & 527 & 608 & 527 & 368 & 267 & 247 \\ 93 & 111 & 208 & 368 & 449 & 368 & 208 & 111 & 93 \\ 9 & 22 & 111 & 267 & 349 & 267 & 111 & 22 & 9 \\ 0 & 9 & 93 & 247 & 327 & 247 & 93 & 9 & 0 \end{bmatrix} \tag{28.128}$$

As to be expected, the interpolation error is zero in the corners of the square, thus for integer coordinates.

Interestingly, the factor k does not vary much for different stochastical models of f, as Tab. 28.2 shows.

d_0	P(Gauss)	P(Lapl)	P(Power)
1.0000	0.3482	0.3710	0.3803
1.5000	0.3425	0.3608	0.3793
2.0000	0.3432	0.3536	0.3789
2.5000	0.3446	0.3490	0.3787
3.0000	0.3459	0.3462	0.3786
3.5000	0.3468	0.3445	0.3785
4.0000	0.3476	0.3437	0.3785

Table 28.2: Ratio k of mean and maximal bi-cubic interpolation error variance for different stochastical models for a signal.

Therefore, we can use the following variances of the model error of the given data, e.g., for g in graylevels $[0, ..., 255]$

$$\boxed{\sigma_{n'_j}^2 = \sigma_{n_j}^2 + k\, \widehat{\mathbb{V}}(J_2 * g) \quad \text{with} \quad k = 0.38\,.} \tag{28.129}$$

depending on the empirical noise variance $\sigma_{n_j}^2$ of the intensities of g_i and the interpolation error derived from the image window.

28.8.2 Noise Variance Estimation

The following section is taken from Förstner (2000)

The noise variance needs to be estimated from images. There are three possible methods to obtain such estimates:

1. *Repeated images*: Taking multiple images of the same scene without changing any parameters yields repeated images. This allows to estimate the noise variance for each individual pixel independently. This certainly is the optimal method in case no model for the noise characteristics is available and can be used as a reference.
   The method is the only one which can handle the case where there is no model for the noise characteristics.
   The disadvantage of this method is the need to have repeated images, which, e. g. in image sequences is difficult to achieve.
2. *Images of homogeneous regions:* Images of homogeneous regions, thus regions with piecewise constant or linear signal, allows to estimate the noise variance from one image alone.

The disadvantage is the requirement for the segmentation of the images into homogeneous regions. Moreover, it is very difficult to guarantee the constancy or linearity of the true intensity image within the homogeneous regions. Small deviations from deficiencies in the illumination already jeopardize this method.
The method is only applicable in case the noise only depends on the signal.

3. *Images with little texture:* Images with a small percentage of textured regions allow to derive the noise variance from the local gradients or curvature. For the larger part of the image they can be assumed to have approximately zero mean. Thus presuming a small percentage of textured regions assumes the expectation of the gradient or the curvature in the homogeneous regions to be negligible compared to the noise.
Also this method is only applicable in case the noise characteristics is only depending on the signal.

We want to describe this method in more detail. We first discuss the method for intensity images. The generalization to range images is straight forward.

28.8.2.1 Estimation of a constant noise variance in intensity Images

The idea is to analyze the histogram of the gradient magnitude of the image in the area where there are no edges and no texture. The procedure given here is similar to that proposed in Förstner (1991).

We now need to specify the model for the ideal image f. We assume that a significant portion $\mathcal{H}$ of the image area $\mathcal{I}$ is homogeneous, thus shows locally constant intensity, thus $\mu_f = const.$. Adopting notions from statistical testing $H_0 = (r,c) \in \mathcal{H}$ is the null-hypothesis, i. e. the hypothesis a pixel belongs to a homogeneous region. Thus

$$\mathsf{E}(\nabla \underline{f} | H_0) = \mathbf{0} \tag{28.130}$$

The other area $\mathcal{I} - \mathcal{H}$ covers edges and textured areas with significantly larger gradients.

Then, as to be shown, the histogram of the homogeneity measure $h = |\nabla g|$ shows exponential behaviour in its left part representing the noise in the image and arbitrary behaviour in the right part representing the edges:

We assume the intensities to be Gaussian distributed with fixed mean and random noise. Assuming the simple gradient kernels

$$\left(\frac{\partial}{\partial r}\right)_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \qquad \left(\frac{\partial}{\partial c}\right)_0 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{28.131}$$

neglecting the scaling factors $1/2$, we obtain the gradient

$$\nabla g = \begin{bmatrix} g_r \\ g_c \end{bmatrix} = \begin{bmatrix} g_{r+1,c} - g_{r-1,c} \\ g_{r,c+1} - g_{r,c-1} \end{bmatrix} \tag{28.132}$$

which is Gaussian distributed with covariance matrix

$$\mathsf{D}\begin{pmatrix} \underline{g}_r \\ \underline{g}_c \end{pmatrix} \Big| H_0 \Big) = \sigma_{n'}^2 \boldsymbol{I} \tag{28.133}$$

Here we use the convention

$$\sigma_{n'}^2 = \sigma_{n_r}^2 = \sigma_{n_c}^2 \tag{28.134}$$

which in general is given by

$$\sigma_{n'}^2 = \int_{x,y} G_{x;s}^2(x,y) dx\, dy = \frac{1}{8\pi s^4}\sigma_n^2, \quad \text{or} \quad \sigma_{n'}^2 = \sum_{r,c} \partial_r^2(r,c)\, \sigma_n^2 \tag{28.135}$$

see Förstner (2000, Eq. (15)). In our case eq. (28.131) leads to

$$\sigma_{n'}^2 = 2\sigma_n^2 \tag{28.136}$$

The squared gradient magnitude measures the homogeneity h

$$h_\nabla(r,c) = |\nabla g(r,c)|^2 = g_r^2(r,c) + g_c^2(r,c) \tag{28.137}$$

It is the sum of two squares of Gaussian variables.

In case the mean $\mu_g = \mu_f$ of $\underline{g}$ is constant in a small region, thus the model eq. (28.130) holds, the squared gradient magnitude is χ_2^2 or exponentially distributed with density function (neglecting the index ∇ for simplicity)

$$p(h|H_0) = \frac{1}{\mu_h} \mathrm{e}^{-\frac{h}{\mu_h}} \tag{28.138}$$

and mean

$$\mathsf{E}(\underline{h}|H_0) = \mu_h = 4\sigma_n^2 \tag{28.139}$$

Therefore, we are able to estimate the parameter μ_h from the empirical density function in the following way:

1. Set the iteration index $\nu = 0$. Specify an approximate value $\sigma_n^{(0)}$ for the noise standard deviation. Use $\mu_h^{(0)} = 4\sigma_n^{2(0)}$ as approximate value for the gradient magnitude.
2. Determine all $h(r,c)$
3. Take the mean $m^{(\nu)}$ of all values $h(r,c) < \mu_h^{(\nu)}$. Its expected value is given by

$$\mu_m^{(\nu)} = \frac{\int_{h=0}^{\mu_h^{(\nu)}} hp(h|H_0)dh}{\int_{h=0}^{\mu_h^{(\nu)}} p(h|H_0)dh} = \frac{\mathrm{e}-2}{\mathrm{e}-1}\,\mu_h^{(\nu)} \tag{28.140}$$

   in case the edges or textured areas do not significantly contribute to this mean. Thus a refined estimate $\mu_h^{(\nu+1)}$ for μ_h is given by:

$$\mu_h^{(\nu+1)} = \frac{\mathrm{e}-1}{\mathrm{e}-2}\, m^{(\nu)} \approx 2.392\, m^{(\nu)} \tag{28.141}$$

4. Set $\nu = \nu + 1$ and repeat step 3.

Usually, only two iterations are necessary to achieve convergence. A modification would be, to take the median of the values $h(r,c)$ as a robust estimate and compensate for the bias caused 1) by taking the median instead of the mean and 2) by the edge pixels (see Brügelmann and Förstner (1992)).

This procedure can be applied to every channel in a multi-channel image, especially in colour images or in gradient images of range images.

28.8.2.2 Estimating a general noise variance function

In case the noise variance is not constant over the whole image area and can be assumed only to depend on the intensity, we need to parameterize the noise variance function $\sigma_n^2 = s(g)$ in some way.

The easiest possibility is to assume it to be continuous. Then we can partition the range $[0..G]$ of all intensities g into intervals $I_\gamma, \gamma = 1..\Gamma$ and assume the noise variance to be constant in each interval.

Thus we repeat the procedure of subsection 28.8.2.1 for each intensity interval under the condition $g \in I_\gamma$.

The choice of the intervals obviously requires some discussion, as it may significantly influence the solution. Taking a set of constant intervals may lead to intervals where no intensities belong to, even in case one would restrict to the real range $[g_{\min}, g_{\max}]$. Therefore, the intervals should be chosen such that

1. they contain enough intensity values.
   The number should be larger than 100 in order to yield precise enough estimates for the noise variances, which in this case has a relative (internal) accuracy better than 10 %. The number of intervals should be chosen in dependency of the expected roughness of $s(g)$. For aerial images we have made good experiences with intervals between 1 and 8 grey values on image patches of 300×300 pixels (cf. Waegli (1998)).
2. they contain an equal number of intensities. This may easily be achieved by using the histogram of the intensities.

28.8.3 Checking the Implementation of the Estimation

This section has been taken from Förstner and Wrobel (2016, Sect. 4.6.8)

Before using the implementation of an estimation procedure we need to check whether it yields correct results. This refers to (1) the estimated parameters, (2) their covariance matrix, and (3) the estimated variance factor. The estimated parameters should be unbiased, the covariance matrix should reflect the sensitivity of the estimated parameters w.r.t. random perturbations of the observations, characterized by the stochastical model, especially the covariance matrix of the observations; and the estimated variance factor should not significantly deviate from 1.

If the implementation is correct, small perturbations in the observations following the stochastical model should lead to small perturbations in the variance factor and in the estimated parameters, where they also should follow the predicted covariance matrix. In the case of larger perturbations, effects of the linearization of a nonlinear model will be visible.

Such an evaluation is based on simulated data, since we then have access to the true values. This also has the advantage that no access to the source code is necessary; the check can be based on the output $\{\widehat{\boldsymbol{\theta}}, \Sigma_{\widehat{\theta}\widehat{\theta}}, \widehat{\sigma_0}^2\}$.

Based on given true values $\tilde{\boldsymbol{\theta}}$ for the parameters, a given observational design, represented by the function $\boldsymbol{f}(\boldsymbol{\theta})$ and a stochastical model $\mathbb{D}(\underline{\boldsymbol{y}}) = \Sigma_{yy}$, we can simulate K samples of observations $\boldsymbol{y}_k$ from

$$\underline{\boldsymbol{y}}_k = \boldsymbol{f}(\tilde{\boldsymbol{\theta}}) - \underline{\boldsymbol{v}}_k\,,\ k = 1, ..., K \qquad \underline{\boldsymbol{v}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{yy})\,, \tag{28.142}$$

leaving the model $\{\boldsymbol{f}(\boldsymbol{\theta}), \Sigma_{yy}\}$ and the true parameters $\tilde{\boldsymbol{\theta}}$ fixed.

The estimation leads to K vectors $\widehat{\boldsymbol{\theta}}_k$ of estimated parameters, to K estimates $\hat{\sigma}_{0k}^2$ of the variance factor, and – provided the relative accuracy $\sigma_y/\mathbb{E}(y)$ of the observations is below 1% – a sufficiently accurate covariance matrix $\Sigma_{\widehat{\theta}\widehat{\theta}}$. In order to be able to check the validity of the model a sufficiently large number K of samples is necessary, which should be larger than the number of elements of the largest covariance matrix which is to be checked.

In the case of Gaussian noise, the evaluation can be based on well-established statistical tests. If one of these tests fails, there are good reasons to doubt whether the program code is a reliable realization of the envisaged estimation model. However, without further tests there are no clues to the source of the discrepancy; it may be the implementation of the envisaged model or of the simulation. This may require more detailed testing.

We now discuss three tests concerning the noise level, the bias, and the validity of the theoretical covariance matrix. They should be performed on a set of representative estimation tasks before using the estimation procedure in a real application.

28.8.3.1 Correctness of the Estimated Noise Level

The correctness of the estimated noise level can be reduced to check the validity of the variance factor. The validity of the estimated variance factor can be based on the mean of the K variance factors derived from the K simulations,

$$s^2 = \frac{1}{K}\sum_{k=1}^{K} \hat{\sigma}_{0k}^2 \,. \tag{28.143}$$

When the implemented model, which is the null hypothesis H_0, holds, the test statistic

$$F = \frac{s^2}{\sigma_0^2}\,, \quad \underline{F}|H_0 \sim F_{KR,\infty} \tag{28.144}$$

is Fisher distributed with KR and ∞ degrees of freedom, where R is the redundancy of the estimation task. If for a specified significance level S, the test statistic $F > F_{KR,\infty;S}$, then the estimated variance factor indicates deviations from the assumed model – possibly caused by implementation errors. In this case, it might be useful to analyse the histogram in order to find possible sources of the deviations.

Observe, this test does not require the theoretical covariance matrix $\Sigma_{\hat{\theta}\hat{\theta}}$ of the estimated parameters.

28.8.3.2 Correctness of the Covariance Matrix

To make sure we can rely on the theoretical covariance matrix provided by the implemented estimation procedure, we compare it with the empirical covariance matrix of the simulation sample. It is given by

$$\hat{\Sigma} = \frac{1}{K-1}\sum_{k=1}^{K} (\hat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{m}}_{\hat{\theta}})(\hat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{m}}_{\hat{\theta}})^{\mathsf{T}} \tag{28.145}$$

with the estimated mean

$$\widehat{\boldsymbol{m}}_{\hat{\theta}} = \frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_k \,. \tag{28.146}$$

When the model holds as implemented and the theoretical precision $\Sigma_{\hat{\theta}\hat{\theta}}$ is correct, the test statistic

$$\underline{X}^2 = (K-1)\left[\ln\left(\det \Sigma_{\hat{\theta}\hat{\theta}} / \det \underline{\hat{\Sigma}}\right) - U + \operatorname{tr}\left(\underline{\hat{\Sigma}}\Sigma_{\hat{\theta}\hat{\theta}}^{-1}\right)\right] \sim \chi^2_{U(U+1)/2} \tag{28.147}$$

is approximately χ^2-distributed with $U(U+1)/2$ degrees of freedom (cf. Koch, 1999, Eq. (2.205)). If for a prespecified significance level S the test statistic X^2 is larger than $\chi^2_{U(U+1)/2,S}$, then there is reason to assume the theoretical covariance matrix, as it results from the implemented model, does not reflect the covariance matrix of $\underline{\hat{\boldsymbol{\theta}}}$ sufficiently well. In this case, it might be useful to visualize the covariance matrix in order to identify possible causes for the found deviation.

It is sufficient to take one of them as reference, though the theoretical covariances of the K samples vary slightly, as the variance propagation is performed not using the true mean, but the estimated parameters. However, as the relative size of this variation is a second-order effect, it can be neglected.

28.8.3.3 Bias in the Estimates

To check the unbiasedness of the estimated parameters we determine their empirical mean.

If the mathematical model holds, the implementation is correct, and higher-order terms during linearization are negligible; the estimated mean of the estimated parameters is Gaussian distributed according to

$$\underline{\widehat{\boldsymbol{m}}}_{\hat{\theta}} \sim \mathcal{N}\left(\tilde{\boldsymbol{\theta}}, \frac{1}{K}\Sigma_{\hat{\theta}\hat{\theta}}\right) \,. \tag{28.148}$$

Under these conditions, the test statistic, the Mahalanobis distance,

$$\underline{X} = K(\widehat{\underline{\boldsymbol{m}}}_{\hat{\theta}} - \tilde{\boldsymbol{\theta}})^{\mathsf{T}} \boldsymbol{\Sigma}_{\hat{\theta}\hat{\theta}}^{-1} (\widehat{\underline{\boldsymbol{m}}}_{\hat{\theta}} - \tilde{\boldsymbol{\theta}}) \sim \chi_U^2 \,, \tag{28.149}$$

is χ^2-distributed with U degrees of freedom. If $X > \chi_{U,S}^2$ for the test statistic and a prespecified significance level S, we have reasons to reject the hypothesis that the model, including the approximations, actually holds as implemented. In this case it might be useful to visualize the bias in order to find possible causes for the rejection of the model.

If these statistical tests are passed on a set of representative simulation data sets, the statistical tests, when applied to real data, can be used as diagnostic tools for identifying discrepancies between the data and the assumed mathematical model.

29 Essential Matrix from Affine Matches and its Precision

Estimating the relative pose of two images, for which the intrinsic camera parameters are known, can be based on affine matches. They contain 6 parameters and 3 of them are needed to estimate the essential matrix. We investigate the uncertainty of the essential matrix based on affine matches.

29.1 Preface

Using affine correspondences promises to speed up the determination of the relative pose (the essential matrix) from a minimum number of correspondences. This is due to the higher number of constraints, namely three, per correspondence, such that only two correspondences are required for a minimal solution of the essential matrix. Since the image area covered by a single affine correspondence is related to the local scale of a keypoint detector, such as Lowe's detector, thus captures only a small part of an image, it appears reasonable to investigate the expected accuracy of the estimated relative pose, i.e., the accuracy of the direction of the basis and the relative rotation. We give an algorithms for the expected accuracy of the relative pose $(\mathbf{b}, R)$. We prove the validity of the algorithm using simulated data. We report on the usefulness of deriving the relative pose from two

affine matches using real data, showing a high variability in the achieved accuracy. Using images of a planar scene, we empirically analyse the difference between the estimated affinities and those derived from the Jacobian of the homography between the images.

On Notation. Homogeneous coordinates and matrices are boldface upright, e.g., coordinates $\mathbf{x}$ or transformation A. Homogeneous coordinates of 2D points ar partitioned $\mathbf{x} = [\boldsymbol{x}_0; x_h]$ into the non-homogeneous part $\boldsymbol{x}_0$ and the homogeneous part x_h. Hence, the non-homogeneous coordinates of the point are $\boldsymbol{x} = \boldsymbol{x}_0/x_h$, in some cases $x_h = 1$. The dimension depends on the context. Stochastical variables are underscored. Observe, we define the derivative of the scalar $c = \boldsymbol{x}^\mathsf{T}\boldsymbol{a}$ w.r.t. a column vector $\boldsymbol{x}$ as a row vector:

$$\frac{\partial \boldsymbol{x}^\mathsf{T}\boldsymbol{a}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{a}^\mathsf{T}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}^\mathsf{T}\,, \tag{29.1}$$

since $\mathrm{d}c = \partial c/\partial \boldsymbol{x}\ \mathrm{d}\boldsymbol{x} = \boldsymbol{a}^\mathsf{T}\mathrm{d}\boldsymbol{x}$.

29.2 The geometry

We start from sensor/pixel coordinates $\boldsymbol{x}$ or $\mathbf{x} = [\boldsymbol{x}; 1]$. For two points $\underline{\mathcal{X}}'$ and $\underline{\mathcal{X}}''$ and the fundamental matrix

$$\mathsf{F} = (\mathsf{K}')^{-\mathsf{T}} R' \mathsf{S}(\mathbf{b}) R''^{\mathsf{T}} (\mathsf{K}'')^{-1} \tag{29.2}$$

we have the constraint

$$\mathbf{x}'^{\mathsf{T}} \mathsf{F} \mathbf{x}'' = 0\,. \tag{29.3}$$

The relative pose is given by the five parameters in $(R, \mathbf{b})$ with $||\mathbf{b}|| = 1$. We now specialize

1. The left camera has rotation $R' = I_3$, the right camera has rotation R.
2. If we know the camera calibration, we can use have camera coordinates, which represent the directions in the camera coordinate system

   $$^c\mathbf{x}' = (\mathsf{K}')^{-1}\mathbf{x}' \quad \text{and} \quad ^c\mathbf{x}'' = (\mathsf{K}'')^{-1}\mathbf{x}'' \tag{29.4}$$

   This is equivalent to

   $$^c\mathbf{x}' \cong \begin{bmatrix} ^ix' \\ ^iy' \\ f' \end{bmatrix} \quad \text{and} \tag{29.5}$$

   The signs will only be relevant when referring to the correct direction of $\mathbf{b}$ and the corresponding rotation R not when referring to the essential or the fundamental matrix.
3. If we know the calibration partially, i.e. up to the focal lengths, we can use image coordinates, which refer to the principle point in a Cartesian image coordinate system

   $$^i\mathbf{x}' = {}^i\mathsf{K}'_c\ {}^c\mathbf{x}' \quad \text{with} \quad {}^i\mathsf{K}'_c = \begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{29.6}$$

4. We further assume, the calibration matrices are identical

   $$\mathsf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \mathsf{Q} \cong \mathsf{K}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f \end{bmatrix}. \tag{29.7}$$

The we have the two epipolar constraints, which we will use for calibrated cameras

$$^c\mathbf{x}'^{\mathsf{T}}\ \mathsf{E}\ {}^c\mathbf{x}'' = 0 \quad \text{with} \quad \mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T} \tag{29.8}$$

or

$$c^{(\mathrm{cal})} = [x_1, x_2, 1] \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ 1 \end{bmatrix} = 0 \tag{29.9}$$

with the Euclideanly normalize camera coordinates

$$x_1 = \frac{{}^c u'}{{}^c w'} = \frac{{}^i x'}{f}\,, \quad x_2 = \frac{{}^c u'}{{}^c w'} = \frac{{}^i y'}{f}\,, \quad z_1 = \frac{{}^c u''}{{}^c w''} = \frac{{}^i x''}{f}\,, \quad z_2 = \frac{{}^c v''}{{}^c w''} = \frac{{}^i y''}{f}\,. \tag{29.10}$$

For partially calibrated cameras we have the constraint

$${}^i\mathbf{x}'^{\mathsf{T}}\, \mathsf{F}\, {}^i\mathbf{x}'' = 0 \quad \text{with} \quad \mathsf{F} = \mathsf{Q}\mathsf{S}(\mathbf{b})R^{\mathsf{T}}\mathsf{Q} \quad \text{and} \quad \mathsf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f \end{bmatrix} \tag{29.11}$$

or

$$c^{(\mathrm{uncal})} = [x_1, x_2, 1] \begin{bmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ 1 \end{bmatrix} \tag{29.12}$$

with the image coordinates

$$x_1 = {}^i x'\,, \quad x_2 = {}^i y'\,, \quad z_1 = {}^i x''\,, \quad \text{and} \quad z_2 = {}^i y''\,. \tag{29.13}$$

In this representation, the constraints differ in the meaning of the Euclideanly normalized coordinates and in the meaning of the 3×3 matrix of the bilinear form. This eases the following derivations.

29.3 Affine Correspondences

29.3.1 Definition of uncertain ACs

Let us assume we observe an affine correspondence (ACs) $(\boldsymbol{x}_0, \boldsymbol{z}_0, \mathsf{A})$. The two points $(\boldsymbol{x}_0, \boldsymbol{z}_0)$ may result from the matching of keypoints. The affinity A may result from a least squares matching.

The local affine transformation between two image patches positioned at $\boldsymbol{x}_0$ in the first image and at $\boldsymbol{z}_0$ in the second image, measured in the image coordinate system, hence assuming calibrated cameras, is given by:

$$\boldsymbol{z} - \boldsymbol{z}_0 = A(\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{b} \quad \text{or} \quad \boldsymbol{z} = \boldsymbol{z}_0 + A(\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{b}\,. \tag{29.14}$$

The affinity refers to the local coordinates $\boldsymbol{x} - \boldsymbol{x}_0$ and in case of perfect correspondence of $(\boldsymbol{x}_0, \boldsymbol{z}_0)$ we have $\boldsymbol{b} = \boldsymbol{0}$.

The affinity A not only contains the local warping of the two images around these points via A but also a local correction $\boldsymbol{b}$ to the position $\boldsymbol{z}_0$ in the second image.

We assume the reference points $(\boldsymbol{x}_0, \boldsymbol{z}_0)$ of the AC are certain, and all uncertainty of the AC is due to the affine transformation A.

$$\underline{\boldsymbol{z}} = (\boldsymbol{z}_0 + \underline{\boldsymbol{b}}) + \underline{A}(\boldsymbol{x} - \boldsymbol{x}_0)\,. \tag{29.15}$$

Any pair $(\boldsymbol{x}, \boldsymbol{z})$ fulfills the model. For simplicity, we chose $\boldsymbol{x} = \boldsymbol{x}_0$ and assume the point pair

$$\{\boldsymbol{x}_0, \boldsymbol{z}_0 + \underline{\boldsymbol{b}}\} \tag{29.16}$$

is corresponding, since the affinity A has no influence on the point in the second image.

We immediately see: the affinity A

$$A = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \tag{29.17}$$

is the Jacobian of $\boldsymbol{z}$ w.r.t $\boldsymbol{x}$. The shift contains the last two parameters of the affinity:

$$\boldsymbol{b} = \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}. \tag{29.18}$$

With the homogeneous matrix

$$\mathsf{A} = \begin{bmatrix} \mathsf{A} & \boldsymbol{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \\ 0 & 0 & 1 \end{bmatrix} = [\mathbf{A}_0 \mid \mathbf{b}_0] \tag{29.19}$$

this can be written with homogeneous coordinates

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{z} = \begin{bmatrix} \boldsymbol{z} \\ 1 \end{bmatrix} \tag{29.20}$$

and the translation matrices

$$\mathsf{T}_x = \begin{bmatrix} \mathsf{I}_2 & \boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \quad \text{and} \quad \mathsf{T}_z = \begin{bmatrix} \mathsf{I}_2 & \boldsymbol{z}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{29.21}$$

and

$$\Delta\mathbf{z} = \mathsf{T}_z^{-1}\mathbf{z} \quad \text{and} \quad \Delta\mathbf{x} = \mathsf{T}_x^{-1}\mathbf{x} \tag{29.22}$$

as

$$\Delta\mathbf{z} = \mathsf{A}\Delta\mathbf{x}\,. \tag{29.23}$$

We assume the matrix A and the covariance matrix Σ_{aa} of the 6 parameters $\boldsymbol{a} = [a_i]$ is provided by the observation process.

Hence, the affine correspondences are given by

$$\boxed{AC: \quad \{\boldsymbol{x}_0, \boldsymbol{z}_0, \mathsf{A}, \Sigma_{aa}\}\,,} \tag{29.24}$$

Thus:

1. the two points $\boldsymbol{x}_0$ and $\boldsymbol{z}_0$ serve as reference points, e.g. are the coordinates of detected keypoints. They are assumed to be fixed, i.e. not stochastic.
2. the two points $\boldsymbol{x}_0$ and $\boldsymbol{z}_0 + \boldsymbol{b}$ are assumed to correspond.
3. the 6 parameters in the homogeneous matrix A are assumed to be observed.
4. the covariance matrix of these 6 parameters is Σ_{aa}.

29.3.2 Coordinate Transformation of Uncertain ACs

We estimate the affinity in the sensor coordinate system, but need it in the camera coordinate system. The relation is

$$\mathbf{x} = \mathsf{K}\,{}^c\mathbf{x} \quad \text{or} \quad {}^c\mathbf{x} = \mathsf{K}^{-1}\,\mathbf{x}\,. \tag{29.25}$$

where we assume Euclideanly normalized homogeneous coordinates. This applies for both images.

It it straight forward to transform the coordinates:

$${}^c\mathbf{x}_0 = \mathsf{K}^{-1}\,\mathbf{x}_0 \quad \text{and} \quad {}^c\mathbf{z}_0 = \mathsf{K}^{-1}\,\mathbf{z}_0\,. \tag{29.26}$$

We now need the matrix A as a function of the matrix

$${}^c\mathsf{A} = \begin{bmatrix} {}^c\mathsf{A} & {}^c\boldsymbol{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{29.27}$$

For this we use the following translation matrices for the camera coordinates

$$ {}^c\mathsf{T}_x = \begin{bmatrix} I_2 & {}^c\boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \quad \text{and} \quad {}^c\mathsf{T}_z = \begin{bmatrix} I_2 & {}^c\boldsymbol{z}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{29.28} $$

with

$$ \mathbf{x}_0 = \mathsf{K}^c\mathbf{x}_0 \quad \text{and} \quad \mathbf{z}_0 = \mathsf{K}^c\mathbf{z}_0 \,. \tag{29.29} $$

Then (29.23) can be written as

$$ \mathsf{T}_z^{-1}\mathbf{z} = \mathsf{A}\mathsf{T}_x^{-1}\mathbf{x} \,. \tag{29.30} $$

Therefore

$$ \mathsf{T}_z^{-1}\mathsf{K}\,{}^c\mathbf{z} = \mathsf{A}\mathsf{T}_x^{-1}\mathsf{K}\,{}^c\mathbf{x} \,. \tag{29.31} $$

or

$$ {}^c\mathbf{z} = \mathsf{K}^{-1}\mathsf{T}_z\mathsf{A}\mathsf{T}_x^{-1}\mathsf{K}\,{}^c\mathbf{x} \,. \tag{29.32} $$

Hence referring to the centred coordinates

$$ {}^c\mathsf{T}_x^{-1\,c}\mathbf{x} \quad \text{and} \quad {}^c\mathsf{T}_z^{-1\,c}\mathbf{z} \tag{29.33} $$

we obtain

$$ {}^c\mathsf{T}_z^{-1\,c}\mathbf{z} = \underbrace{{}^c\mathsf{T}_z^{-1}\mathsf{K}^{-1}\mathsf{T}_z\mathsf{A}\mathsf{T}_x^{-1}\mathsf{K}\,({}^c\mathsf{T}_x^{-1})^{-1}}_{{}^c\mathsf{A}}\,{}^c\mathsf{T}_x^{-1\,c}\mathbf{x} \,. \tag{29.34} $$

Thus, finally, with

$$ {}^c\mathsf{A} = {}^c\mathsf{T}_z^{-1}\mathsf{K}^{-1}\mathsf{T}_z\mathsf{A}\mathsf{T}_x^{-1}\mathsf{K}\,{}^c\mathsf{T}_x \tag{29.35} $$

we have the local affinity in the camera coordinate system equivalent to (29.30)

$$ {}^c\mathsf{T}_z^{-1\,c}\mathbf{z} = {}^c\mathsf{A}\,{}^c\mathsf{T}_x^{-1\,c}\mathbf{x} \,. \tag{29.36} $$

Now, using $\boldsymbol{x}_0 = K^c\boldsymbol{x}_0 + \boldsymbol{k}$ we observe

$$ \begin{aligned} \mathsf{T}_x^{-1}\mathsf{K}\,{}^c\mathsf{T}_x &= \begin{bmatrix} I_2 & -\boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} K & \boldsymbol{k} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} I_2 & {}^c\boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (29.37) \\ &= \begin{bmatrix} K & \boldsymbol{k} - \boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} I_2 & {}^c\boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (29.38) \\ &= \begin{bmatrix} K & K^c\boldsymbol{x}_0 + (\boldsymbol{k} - \boldsymbol{x}_0) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (29.39) \\ &= \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} && (29.40) \end{aligned} $$

for arbitrary $\boldsymbol{x}_0$. Therefore we have the simple relation

$$ {}^c\mathsf{A} = \begin{bmatrix} K^{-1} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} A & \boldsymbol{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{29.41} $$

or

$$ {}^c\mathsf{A} = \begin{bmatrix} K^{-1}AK & K^{-1}\boldsymbol{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{29.42} $$

The covariance matrix of the affinity in the camera coordinate system can be derived from

$$ {}^c\mathsf{A} = \mathsf{K}_0^{-1}\mathsf{A}\mathsf{K}_0 \quad \text{with} \quad \mathsf{K}_0 = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{29.43} $$

Vectorization yields

$$ \mathrm{vec}({}^c\mathsf{A}) = (\mathsf{K}_0^\mathsf{T} \otimes \mathsf{K}_0^{-1})\mathrm{vec}\mathsf{A} \,. \tag{29.44} $$

from which we obtain the covariance matrix of the affinity in the camera system

$$ \Sigma_{{}^c\mathrm{a}{}^c\mathrm{a}} = (\mathsf{K}_0^\mathsf{T} \otimes \mathsf{K}_0^{-1})\Sigma_{\mathrm{aa}}(\mathsf{K}_0^\mathsf{T} \otimes \mathsf{K}_0^{-1})^\mathsf{T} \,, \tag{29.45} $$

– except for some zero rows and columns in the covariance matrices, which we add and delete in order to start with a 9×9 covariance matrix Σ_{aa} and finally obtain a regular 6×6 covariance matrix $\Sigma_{{}^c a{}^c a}$.

29.3.3 Theoretical Precision of ACs

The covariance matrix of an AC can be derived from the normal equation matrix of the least squares matching. We use the approximate template matching for the analysis of the theoretical precision The normal equation matrix for LSM with a 6-parameter affinity is:

$$N = \frac{1}{\sigma_n^2} \sum_{ij} \nabla f_\theta(i,j) \nabla^\mathsf{T} f_\theta(i,j) \tag{29.46}$$

with

$$\nabla f_\theta(i,j) = \begin{bmatrix} if_i \\ if_j \\ jf_i \\ jf_j \\ f_i \\ f_j \end{bmatrix}, \tag{29.47}$$

and the image noise variance σ_n^2, which is assumed to be the same for all pixels. Since we do not know f, let us assume the following:

- The expectation of $\underline{f}$ is 0.
- The variance of f_i and f_j are $\sigma_{f'}^2$. Their correlation is 0.

Then the normal equation matrix for a $K \times K$ image is diagonal with the entries

$$N = \frac{\sigma_f'^2}{\sigma_n^2} \mathrm{Diag}([s,s,s,s,t,t]) \quad \text{with} \quad s = \frac{1}{12} K^2(K^2-1) \quad \text{and} \quad t = K^2\,. \tag{29.48}$$

Hence the precision, i.e. the inverse standard deviations of the shifts and the affine parameters increase with the scale or the window size. In detail, for some constant

$$\kappa = \frac{\sigma_n}{\sigma_{f'}} \tag{29.49}$$

the standard deviations of the 6 parameters are approximately related to the diameter K of the image patch by:

$$\sigma_{\theta_u} = \sqrt{12}\,\kappa\,\frac{1}{K^2} \text{ for } u = 1,2,3,4 \quad \text{and} \quad \sigma_{\theta_u} = \kappa\,\frac{1}{K} \text{ for } u = 5,6\,. \tag{29.50}$$

Since the precondition for this relation is that the variance of the gradient is the same and homogeneous over all the image windows, this theoretical relation will only approximately hold for real images.

However, we may use this result to generate reasonable artificial covariance matrices:

$$\Sigma_{aa} = \begin{bmatrix} \sigma_a^2 I_4 & 0 \\ 0 & \sigma_b^2 I_2 \end{bmatrix} \quad \text{with} \quad \sigma_a = \frac{\sqrt{12}}{K^2}\frac{1}{\sigma_{f'}}\sigma_n \quad \text{and} \quad \sigma_b = \frac{1}{K}\frac{1}{\sigma_{f'}}\sigma_n \tag{29.51}$$

for some $\kappa = \sigma_n/\sigma_{f'}$ and using the assumed side length K of the image window.

In some of my experiments I used $\sigma_n = 1$ [gr] and $\sigma_{f'} = 5$ [gr/pixel] hence, the constant $\kappa = 1/5$ [pixel]. Normally, we assume the pixel distance to be 1, unitless.

29.4 Precision of relative orientation from pairs of affine matches

29.4.1 The constraints

29.4.1.1 Point constraints

Since we have 4 observed image coordinates but 3 unknown 3D coordinates, there is one constraint - the epipolar constraint between the image coordinates.

If the two poses are given by

$$\mathsf{P}_2 = [I_3 \mid \mathbf{0}] \quad \text{and} \quad P_2 = [R \mid \mathbf{b}] \tag{29.52}$$

then, given a corresponding point pair with camera coordinates[1] $(\mathbf{x}, \mathbf{z}) := ({}^c\mathbf{x}', {}^c\mathbf{x}'')$ the epipolar constraint for the essential matrix E is given by

$$c(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\mathsf{T}\mathsf{E}\mathbf{z} = 0 \quad \text{with} \quad \mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T} = \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix}. \tag{29.53}$$

or with the Euclideanly normalized homogeneous coordinates $\mathbf{x} := [\boldsymbol{x}; 1]$ and $\mathbf{z} := [\boldsymbol{z}; 1]$

$$c = (\mathbf{z}^\mathsf{T} \otimes \mathbf{x}^\mathsf{T})\,\mathbf{e} = ([\boldsymbol{z}^\mathsf{T}, 1] \otimes [\boldsymbol{x}^\mathsf{T}, 1])\,\mathbf{e} = 0 \quad \text{with} \quad \mathbf{e} = \mathrm{vec}\mathsf{E}\,, \tag{29.54}$$

or explicitly

$$c = [x_1 z_1 \; x_2 z_1 \; z_1 \; x_1 z_2 \; x_2 z_2 \; z_2 \; x_1 \; x_2 \; 1]\,\mathbf{e} = 0 \tag{29.55}$$

Hence, with 5 corresponding points we may derive the 5 parameters of the relative pose. Thus, the coordinates $\boldsymbol{x}$ and $\boldsymbol{z}$ can be assumed to be observed in a calibrated camera with focal length $f = 1$.

29.4.1.2 Affinity constraints

Since we have 6 observed parameters but 3 unknown parameters of the (locally) planar surface, there are 3 constraints between the 6 parameters. One expresses the epipolar constraint of the two points $\boldsymbol{x}_0$ and $\boldsymbol{y}$ from (29.14). The other two relate the 4 affine parameters $(a_1, ..., a_4)$ to the 2 parameters of the surface slope.

In order to arrive at constraints for the affinity we may differentiate (29.53) w.r.t $\boldsymbol{x}$, leading to the 2 constraints

$$\frac{\partial \mathbf{x}^\mathsf{T}\mathsf{E}\mathbf{z}}{\partial \boldsymbol{x}} = \frac{\partial \mathbf{z}^\mathsf{T}\mathsf{E}^\mathsf{T}\mathbf{x}}{\partial \boldsymbol{x}} = \mathbf{0} \tag{29.56}$$

Since e have

$$\frac{\partial \mathbf{x}^\mathsf{T}}{\partial \boldsymbol{x}} = [I_2 \mid \mathbf{0}] \quad \text{and} \quad \frac{\partial \mathbf{z}^\mathsf{T}}{\partial \boldsymbol{x}} = [A^\mathsf{T} \mid \mathbf{0}] \tag{29.57}$$

we obtain

$$\underbrace{\frac{\partial \mathbf{x}^\mathsf{T}}{\partial \boldsymbol{x}}}_{2\times 3}\,\mathsf{E}\,\mathbf{z} + \underbrace{\frac{\partial \mathbf{z}^\mathsf{T}}{\partial \boldsymbol{x}}}_{2\times 3}\,\mathsf{E}^\mathsf{T}\,\mathbf{x} = \mathbf{0}\,. \tag{29.58}$$

With (29.20) this is given by (see Eichhardt and Chetverikov (2018, eq. (8))

$$[I_2 \mid \mathbf{0}]\,\mathsf{E}\,\mathbf{z} + [A^\mathsf{T} \mid \mathbf{0}]\,\mathsf{E}^\mathsf{T}\,\mathbf{x} = \mathbf{0}\,. \tag{29.59}$$

If we include twice the epipolar point constraint (29.53) we obtain the 3 combined constraints

$$\boxed{\boldsymbol{c} = \begin{bmatrix} I_2 & \boldsymbol{x} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}^\mathsf{T} \mathsf{E}\boldsymbol{z} + \begin{bmatrix} A & \boldsymbol{z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}^\mathsf{T} \mathsf{E}^\mathsf{T}\mathbf{x} = \mathbf{0}\,.} \tag{29.60}$$

For simplifying notation we introduce the matrices

$$X = \begin{bmatrix} I_2 & \boldsymbol{x}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} =: [\mathsf{P}_0 \mid \mathbf{x}_0] \quad \text{and} \quad Z = \begin{bmatrix} A & \boldsymbol{z} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} =: [\mathsf{A}_0 \mid \mathbf{z}]\,. \tag{29.61}$$

[1] We choose the name z for the points in the right image, since we use the letter y for the observations in the estimation process.

and obtain the combined constraints

$$\boxed{\boldsymbol{c} = X^\mathsf{T}\mathsf{E}\mathbf{z} + Z^\mathsf{T}\mathsf{E}^\mathsf{T}\mathbf{x} = \mathbf{0}\,.} \tag{29.62}$$

We alternatively may differentiate (29.54) w.r.t $\boldsymbol{x}$ to obtain

$$\frac{\partial c}{\partial \boldsymbol{x}} = \left(\mathbf{z}^\mathsf{T} \otimes [I_2|\mathbf{0}] + [A^\mathsf{T}|\mathbf{0}] \otimes \mathbf{x}^\mathsf{T}\right)\, \mathbf{e} \tag{29.63}$$

Together we obtain per pair the three constraints

$$\boldsymbol{c} = \left(\mathbf{z}^\mathsf{T} \otimes \begin{bmatrix} I_2 & \mathbf{0} \\ \boldsymbol{x}^\mathsf{T} & 1 \end{bmatrix} + \begin{bmatrix} A^\mathsf{T} & \mathbf{0} \\ \boldsymbol{z}^\mathsf{T} & 1 \end{bmatrix} \otimes \mathbf{x}^\mathsf{T}\right)\, \mathbf{e} = \mathbf{0}\,, \tag{29.64}$$

or

$$\boldsymbol{c} = \left(\mathbf{z}^\mathsf{T} \otimes X^\mathsf{T} + Z^\mathsf{T} \otimes \mathbf{x}^\mathsf{T}\right)\, \mathbf{e} = \mathbf{0}\,, \tag{29.65}$$

or

$$\boldsymbol{c} = C^\mathsf{T}\mathbf{e} = \mathbf{0} \tag{29.66}$$

with the 3×9 matrix

$$C = C_1 + C_2 \tag{29.67}$$

with

$$C_1 = [z_1 X^\mathsf{T}, z_2 X^\mathsf{T}, X^\mathsf{T}] = \begin{bmatrix} z_1 & 0 & 0 & z_2 & 0 & 0 & 1 & 0 & 0 \\ 0 & z_1 & 0 & 0 & z_2 & 0 & 0 & 1 & 0 \\ x_1 z_1 & x_2 z_1 & z_1 & x_1 z_2 & x_2 z_2 & z_2 & x_1 & x_2 & 1 \end{bmatrix} \tag{29.68}$$

and

$$C_2 = \begin{bmatrix} a_1\boldsymbol{x}^\mathsf{T} & a_2\boldsymbol{x}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ a_3\boldsymbol{x}^\mathsf{T} & a_4\boldsymbol{x}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ z_1\boldsymbol{x}^\mathsf{T} & z_2\boldsymbol{x}^\mathsf{T} & \boldsymbol{x}^\mathsf{T} \end{bmatrix} = \begin{bmatrix} a_1 x_1 & a_1 x_2 & a_1 & a_2 x_1 & a_2 x_2 & a_1 & 0 & 0 & 0 \\ a_3 x_1 & a_3 x_2 & a_3 & a_4 x_1 & a_4 x_2 & a_4 & 0 & 0 & 0 \\ x_1 z_1 & x_2 z_1 & z_1 & x_1 z_2 & x_2 z_2 & z_2 & x_1 & x_2 & 1 \end{bmatrix} \tag{29.69}$$

$$C^\mathsf{T} = \begin{bmatrix} z_1 + a_1 x_1 & a_1 x_2 & a_1 & z_2 + a_2 x_1 & a_2 x_2 & a_1 & 1 & 0 & 0 \\ a_3 x_1 & z_1 + a_3 x_2 & a_3 & a_4 x_1 & z_2 + a_4 x_2 & a_4 & 0 & 1 & 0 \\ 2x_1 z_1 & 2x_2 z_1 & 2z_1 & 2x_1 z_2 & 2x_2 z_2 & 2z_2 & 2x_1 & 2x_2 & 2 \end{bmatrix}$$

Remark: Some authors perform the vectorization of the epipolar constraint based on its transposed. So, starting from

$$\mathbf{z}^\mathsf{T}\mathsf{E}^\mathsf{T}\mathbf{x} = 0 \tag{29.70}$$

they arrive at

$$(\mathbf{x}^\mathsf{T} \otimes \mathbf{z})\overline{\mathbf{e}} = 0 \quad \text{with} \quad \overline{\mathbf{e}} = \text{vec}\left(\mathsf{E}^\mathsf{T}\right) \tag{29.71}$$

i.e., using a row-wise vectorization of E. This obviously leads to expressions where the two factors of the Kronecker products are exchanged. This leads to the same constraints, however, the columns of its entries permuted.

For the combined constraints we obtain

$$\boldsymbol{c} = \left(X^\mathsf{T} \otimes \mathbf{z}^\mathsf{T} + \mathbf{x}^\mathsf{T} \otimes Z^\mathsf{T}\right)\, \overline{\mathbf{e}} = \mathbf{0}\,, \tag{29.72}$$

and

$$\overline{C}^\mathsf{T}\overline{\mathbf{e}} = \mathbf{0} \tag{29.73}$$

with

$$\overline{C}^\mathsf{T} = \begin{bmatrix} z_1 + a_1 x_1 & z_2 + a_2 x_1 & 1 & a_1 x_2 & a_2 x_2 & 0 & a_1 & a_2 & 0 \\ a_3 x_1 & a_4 x_1 & 0 & z_1 + a_3 x_2 & z_2 + a_4 x_2 & 1 & a_3 & a_4 & 0 \\ 2x_1 z_1 & 2x_1 z_2 & 2x_1 & 2x_2 z_1 & 2x_2 z_2 & 2x_2 & 2z_1 & 2z_2 & 2 \end{bmatrix}$$

This directly relates to Eichhardt and Chetverikov (2018, eqs. (15), (16)), taking into account that they refer to a row-wise numbering of the elements of A, hence, a_2 and a_3 are exchanged compared to our definition. ◇

29.4.2 Relative pose from two affine correspondences

With two AC's we obtain 6 constraints for the 5 parameters. This allows us to estimate the essential matrix by approximating the 6×9 matrix $\mathsf{C}^\mathsf{T}(\boldsymbol{y}) = [\mathsf{C}_1(\boldsymbol{y}_1), \mathsf{C}_2(\boldsymbol{y}_2)]^\mathsf{T}$ with a rank 5-matrix and using its 4-dimensional nullspace for estimating E, similar to the 5-point algorithm, since the 6×9 matrix C^T generally has rank 6.

This approximation can be done in two ways:

1. If its reduced SVD is given by

$$\underset{6\times 6}{\mathsf{C}^\mathsf{T}} = \underset{6\times 6}{\mathsf{U}}\ \underset{6\times 6}{\mathsf{D}}\ \underset{6\times 9}{\mathsf{V}^\mathsf{T}} \tag{29.74}$$

with

$$\underset{6\times 1}{\mathsf{U}} = [\underset{6\times 1}{\boldsymbol{u}_1}, \ldots, \boldsymbol{u}_i, \ldots, \boldsymbol{u}_6]\,, \quad \mathsf{D} = \mathrm{Diag}([d_i]) \quad \text{and} \quad \underset{9\times 1}{\mathsf{V}} = [\underset{9\times 1}{\boldsymbol{v}_1}, \ldots, \boldsymbol{v}_i, \ldots, \boldsymbol{v}_6] \tag{29.75}$$

then the rank-5 approximation $\mathsf{C}_\rho^\mathsf{T}$ of C^T is

$$\boldsymbol{\rho}(\mathsf{C}^\mathsf{T}) =: \mathsf{C}_\rho^\mathsf{T} = \sum_{i=1}^{5} d_i\, \boldsymbol{u}_i\, \boldsymbol{v}_i^\mathsf{T} = \mathsf{C}^\mathsf{T} - d_6\, \boldsymbol{u}_6\, \boldsymbol{v}_6^\mathsf{T}\,. \tag{29.76}$$

Actually the matrix $\mathsf{C}_\rho^\mathsf{T}$ is used for the determination of E. Hence, for determining the covariance matrix of the pose, we need the Jacobians of the modified constraints

$$\mathsf{C}_\rho^\mathsf{T}(\boldsymbol{y}, \boldsymbol{\theta})\, \bar{\mathbf{e}}(\boldsymbol{\theta}) = \mathbf{0}\,. \tag{29.77}$$

2. Alternatively, we may omit one of the 6 constraints, thus eliminate one row $\boldsymbol{c}_k$ of C^T. The choice may again use the SVD: we choose that k, where the angle between $\boldsymbol{c}_k$ and $\boldsymbol{v}_6$ is smallest. We have the cos of the rows of C^T with $\boldsymbol{v}_6$:

$$\cos(c_k, \boldsymbol{v}_6) = \mathsf{U}\mathsf{D}\mathsf{V}^\mathsf{T}\boldsymbol{v}_6/||\boldsymbol{c}_k|| = \mathsf{U}\mathsf{D}\boldsymbol{e}_6/||\boldsymbol{c}_k|| = d_6 \mathsf{U}_{:,6}/||\boldsymbol{c}_k||\,. \tag{29.78}$$

Hence, we choose that k for which the cosine is largest. Hence, we work with the 5×9 matrix

$$\mathsf{C}_r^\mathsf{T} = \mathsf{C}^{[k]\mathsf{T}}\,. \tag{29.79}$$

We only discuss the second option, since we do not have the Jacobian for the SVD of a rectangular matrix.

29.4.3 Uncertainty of the relative pose with selected constraints

In the following we assume the two points $(\boldsymbol{x}_0, \boldsymbol{z}_0)$ are certain and the complete uncertainty of the affine matches is contained in the 6 parameters $\boldsymbol{\theta} = [a_1, ..., a_6]^\mathsf{T}$ of A. Further, we assume the covariance matrix Σ_{yy} is available from the matching process. With the observations $\boldsymbol{y} = [a_1, ..., a_6]^\mathsf{T}$ and the unknown parameters $\boldsymbol{\theta} \equiv (R, \mathbf{b})$ the constraints (29.60) have the form $r(\boldsymbol{C}(\boldsymbol{y}, \boldsymbol{\theta}))\, \bar{\mathbf{e}} = \mathbf{0}$ or explicitly

$$\boldsymbol{c}_r(\boldsymbol{y}, \boldsymbol{\theta}) = \boldsymbol{r}\left(\begin{bmatrix} \mathsf{X}_1^\mathsf{T} \otimes \mathbf{z}_1^\mathsf{T} + \mathbf{x}_1^\mathsf{T} \otimes \mathsf{Y}_1^\mathsf{T} \\ \mathsf{X}_2^\mathsf{T} \otimes \mathbf{z}_2^\mathsf{T} + \mathbf{x}_2^\mathsf{T} \otimes \mathsf{Y}_2^\mathsf{T} \end{bmatrix}\right) \bar{\mathbf{e}} = \mathsf{C}_r^\mathsf{T}\, \bar{\mathbf{e}} = \mathbf{0} \tag{29.80}$$

If the covariance matrix Σ_{yy} of the given observations $\boldsymbol{y}$ is given, we can derive the covariance matrix $\Sigma_{\theta\theta}$ of the estimated 5 parameters of the relative pose using the two Jacobians

$$\mathsf{B}_r^\mathsf{T} = \frac{\partial \boldsymbol{c}_r}{\partial \boldsymbol{y}} \quad \text{and} \quad \mathsf{D}_r = \frac{\partial \boldsymbol{c}_r}{\partial \boldsymbol{\theta}} \tag{29.81}$$

from

$$\Sigma_{\hat{\theta}\hat{\theta}} = \left(\mathsf{D}_r^\mathsf{T}(\mathsf{B}_r^\mathsf{T}\Sigma_{yy}\mathsf{B}_r)^{-1}\mathsf{D}_r\right)^{-1} \overset{\text{if } |\mathsf{D}_r| \neq 0}{=} \mathsf{D}_r^{-1}\, \mathsf{B}_r^\mathsf{T}\Sigma_{yy}\mathsf{B}_r\mathsf{D}_r^{-\mathsf{T}}\,. \tag{29.82}$$

The derivation can be generalized for observed directions $\mathbf{x}^e = \mathbf{x}/||\mathbf{x}||$ and $\mathbf{z}^e = \mathbf{z}/||\mathbf{z}||$, when observing the local affinity in the tangent space of $\mathbf{x}^e$.

29.4.4 The Jacobians for selected constraints

We now want to derive the Jacobians of the constraints w.r.t the 12 observed affine parameters $\boldsymbol{y}$ in

$$\mathsf{A}_1 = \begin{bmatrix} y_1 & y_3 & y_5 \\ y_2 & y_4 & y_6 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathsf{A}_2 = \begin{bmatrix} y_7 & y_9 & y_{11} \\ y_8 & y_{10} & y_{12} \\ 0 & 0 & 1 \end{bmatrix} \tag{29.83}$$

and 5 parameters of the relative pose, still to be specified.

The derivatives for the selected constraints w.r.t. the observations follow from

$$\begin{aligned} \underset{5\times 12}{B_r^\mathsf{T}} &= \begin{bmatrix} \frac{\partial c_{r_1}}{\partial \boldsymbol{y}} \\ \cdots \\ \frac{\partial c_{r_k}}{\partial \boldsymbol{y}} \\ \cdots \\ \frac{\partial c_{r_5}}{\partial \boldsymbol{y}} \end{bmatrix} && (29.84) \\ &= \begin{bmatrix} \frac{\partial (\boldsymbol{c}_{rk}^\mathsf{T} \overline{\mathbf{e}})}{\partial \boldsymbol{y}} \end{bmatrix} && (29.85) \\ &= \begin{bmatrix} \frac{\partial (\overline{\mathbf{e}}^\mathsf{T} \boldsymbol{c}_{r_k})}{\partial \boldsymbol{y}} \end{bmatrix} && (29.86) \\ &= \begin{bmatrix} \overline{\mathbf{e}}^\mathsf{T} \underbrace{\frac{\partial \boldsymbol{c}_{r_k})}{\partial \boldsymbol{y}}}_{9\times 12} \end{bmatrix} && (29.87) \\ &= \begin{bmatrix} \overline{\mathbf{e}}^\mathsf{T} \underbrace{\frac{\partial \boldsymbol{c}_{r_k}(C^\mathsf{T})}{\partial \mathrm{vec} C}}_{9\times 54} \underbrace{\frac{\partial \mathrm{vec} C(\boldsymbol{y})}{\partial \boldsymbol{y}}}_{54\times 12} \end{bmatrix} && (29.88) \end{aligned}$$

Observe, vec$C$ is a 54-vector.

Similarly we need the Jacobian w.r.t. parameters $\boldsymbol{\theta}$

$$D^\mathsf{T} = \frac{\partial \boldsymbol{c}_r}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{c}_r}{\partial \overline{\mathbf{e}}} \frac{\partial \overline{\mathbf{e}}}{\partial \boldsymbol{\theta}} = C_r^\mathsf{T} \frac{\partial \overline{\mathbf{e}}}{\partial \boldsymbol{\theta}} \tag{29.89}$$

We start with

- determining he derivatives of vec$C = [\boldsymbol{c}_1; \boldsymbol{c}_2; \boldsymbol{c}_3]$ w.r.t. the observations $\boldsymbol{y}$, this can be achieved separately for each AC. The selection is done separately.
- determine the Jacobian of the elements $\overline{\mathbf{e}}$ of the essential matrix w.r.t. pose parameters $\boldsymbol{\theta}$, this is independent of the constraint matrix; and finally
- addressing the Jacobian of the selected constraints $\partial \boldsymbol{c}_r(C^\mathsf{T})/\partial(\mathrm{vec} C)$.

29.4.4.1 The Jacobians of C^T w.r.t. observations y

We can determine the Jacobian for each AC separately.

With approximate values $\boldsymbol{x} = \boldsymbol{x}_0$ have from (29.15)

$$\mathrm{d}\boldsymbol{z} = \mathrm{d}\boldsymbol{b} = \mathrm{d} \begin{bmatrix} y_5 \\ y_6 \end{bmatrix} \quad \text{and} \quad \mathrm{d}A = \mathrm{d} \begin{bmatrix} y_1 & y_3 \\ y_2 & y_4 \end{bmatrix} \tag{29.90}$$

and

$$\mathrm{d}\mathbf{z} = \mathrm{d} \begin{bmatrix} y_5 \\ y_6 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathrm{d}A_0 = \mathrm{d} \begin{bmatrix} y_1 & y_3 \\ y_2 & y_4 \\ 0 & 0 \end{bmatrix}. \tag{29.91}$$

We use the 3×6 matrices

$$Y_i^\mathsf{T} = \boldsymbol{e}_i^\mathsf{T} \otimes \mathsf{P}_0 \quad \text{e.g.} \quad Y_1^\mathsf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{29.92}$$

Then we can express the derivatives as

$$\mathrm{d}\mathbf{z} = Y_3^\mathsf{T} \mathrm{d}\boldsymbol{y} \quad \text{and} \quad \mathrm{d}A_0 = \left[Y_1^\mathsf{T} \mathrm{d}\boldsymbol{y} \mid Y_2^\mathsf{T} \mathrm{d}\boldsymbol{y} \right] \tag{29.93}$$

The three rows of the matrix C^T from (29.166)

$$\begin{bmatrix} \boldsymbol{c}_1^\mathsf{T} \\ \boldsymbol{c}_2^\mathsf{T} \\ \boldsymbol{c}_3^\mathsf{T} \end{bmatrix} = C^\mathsf{T} = X^\mathsf{T} \otimes \mathbf{z}^\mathsf{T}(\boldsymbol{y}) + \mathbf{x}^\mathsf{T} \otimes Z^\mathsf{T}(\boldsymbol{y}) \tag{29.94}$$

are

$$\begin{bmatrix} \boldsymbol{c}_1^\mathsf{T} \\ \boldsymbol{c}_2^\mathsf{T} \\ \boldsymbol{c}_3^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1^\mathsf{T} \otimes \mathbf{z}^\mathsf{T}(\boldsymbol{y}) + \mathbf{x}^\mathsf{T} \otimes [y_1, y_2, 0] \\ \boldsymbol{e}_2^\mathsf{T} \otimes \mathbf{z}^\mathsf{T}(\boldsymbol{y}) + \mathbf{x}^\mathsf{T} \otimes [y_3, y_4, 0] \\ 2\mathbf{x}^\mathsf{T} \otimes \mathbf{z}^\mathsf{T}(\boldsymbol{y}) \end{bmatrix} \tag{29.95}$$

For the vec-product we have e.g.

$$\underset{9\times 6}{\frac{\partial \boldsymbol{x} \otimes \mathbf{z}(\boldsymbol{y})}{\partial \boldsymbol{y}}} = \frac{\partial \begin{bmatrix} x_1 \mathbf{z}(\boldsymbol{y}) \\ x_2 \mathbf{z}(\boldsymbol{y}) \\ x_3 \mathbf{z}(\boldsymbol{y}) \end{bmatrix}}{\partial \boldsymbol{y}} = \begin{bmatrix} x_1 Y_3^\mathsf{T} \\ x_2 Y_3^\mathsf{T} \\ x_3 Y_3^\mathsf{T} \end{bmatrix} = \mathbf{x} \otimes Y_3 \,. \tag{29.96}$$

Hence we obtain the Jacobians

$$\underbrace{\frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{y}}}_{9\times 6} = \boldsymbol{e}_1 \otimes Y_3^\mathsf{T} + \mathbf{x} \otimes Y_1^\mathsf{T} \tag{29.97}$$

$$\frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{y}} = \boldsymbol{e}_2 \otimes Y_3^\mathsf{T} + \mathbf{x} \otimes Y_2 \tag{29.98}$$

$$\frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{y}} = 2\mathbf{x} \otimes Y_3^\mathsf{T} \,. \tag{29.99}$$

or simply for each AC

$$\underset{27\times 6}{\frac{\partial \mathrm{vec}\, C_i}{\partial \boldsymbol{y}}} = \begin{bmatrix} \boldsymbol{e}_1 \otimes Y_3^\mathsf{T} + \mathbf{x}_i \otimes Y_1^\mathsf{T} \\ \boldsymbol{e}_2 \otimes Y_3^\mathsf{T} + \mathbf{x}_i \otimes Y_2 \\ 2\mathbf{x}_i \otimes Y_3^\mathsf{T} \end{bmatrix}. \tag{29.100}$$

therfore

$$\boxed{\underset{54\times 12}{\frac{\partial \mathrm{vec}\, C}{\partial \boldsymbol{y}}} = \begin{bmatrix} \frac{\partial \mathrm{vec}\, C_1}{\partial \boldsymbol{y}} & 0 \\ 0 & \frac{\partial \mathrm{vec}\, C_2}{\partial \boldsymbol{y}} \end{bmatrix}.} \tag{29.101}$$

29.4.4.2 The Jacobians of $\bar{\mathbf{e}}$ w.r.t. parameters $\boldsymbol{\theta}$

We represent the uncertain rotation matrix via the product of the mean rotation μ_R and an uncertain small rotation $\exp(S(\underline{\boldsymbol{r}}))$

$$\underline{R} = \exp(S(\underline{\boldsymbol{r}}))\mu_R \approx \mu_R + S(\mathrm{d}\boldsymbol{r})\mu_R \,. \tag{29.102}$$

Similarly we represent the uncertain basis vector $\underline{\mathbf{b}}$ via an uncertain 2-vector $\underline{\boldsymbol{b}}_r$ perpendicular to $\boldsymbol{\mu}_b$:

$$\underline{\mathbf{b}} = \mathsf{N}(\boldsymbol{\mu}_b + J_r(\boldsymbol{\mu}_b)\underline{\boldsymbol{b}}_r) \quad \text{with} \quad \underset{3\times 2}{J_r(\boldsymbol{\mu}_b)} = \mathsf{null}(\boldsymbol{\mu}_b^{\mathsf{T}})\,. \tag{29.103}$$

We therefore have

$$\mathrm{d}\mathbf{b} = J_r\,\mathrm{d}\boldsymbol{b}_r \quad \text{and} \quad \mathrm{d}R = S(\mathrm{d}\boldsymbol{r})R\,, \tag{29.104}$$

since $||\mathbf{b}|| = 1$ and assuming $R = \mu_R$. We collect the uncertain five parameters in the vector

$$\underline{\boldsymbol{\theta}} = \begin{bmatrix} \underline{\boldsymbol{b}}_r \\ \underline{\boldsymbol{r}} \end{bmatrix} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{\theta}}) = \Sigma_{\theta\theta} \tag{29.105}$$

First we determine the differential of $\mathsf{E} = \mathsf{S}(\mathbf{b})R^{\mathsf{T}}$:

$$\mathrm{d}\mathsf{E}(\boldsymbol{\theta}) = \mathrm{d}\mathsf{S}(\mathbf{b})R^{\mathsf{T}} + \mathsf{S}(\mathbf{b})(S(\mathrm{d}\boldsymbol{r})R)^{\mathsf{T}} = \mathsf{S}(J_r\mathrm{d}\boldsymbol{b}_r)R^{\mathsf{T}} + \mathsf{E}S^{\mathsf{T}}(\mathrm{d}\boldsymbol{r})\,. \tag{29.106}$$

The transposed

$$\mathrm{d}\mathsf{E}^{\mathsf{T}}(\boldsymbol{\theta}) = R\mathsf{S}^{\mathsf{T}}(J_r\mathrm{d}\boldsymbol{b}_r) + S(\mathrm{d}\boldsymbol{r})\mathsf{E}^{\mathsf{T}}\,. \tag{29.107}$$

can be rewritten as

$$\mathrm{d}\overline{\mathbf{e}} = \mathrm{vec}\left(\mathrm{d}\mathsf{E}^{\mathsf{T}}\right) = (I_3 \otimes R)\,\mathrm{vec}\left(\mathsf{S}^{\mathsf{T}}(J_r\mathrm{d}\boldsymbol{b}_r)\right) + (\mathsf{E}\otimes I_3)\,\mathrm{vec}\,(S(\mathrm{d}\boldsymbol{r})) \tag{29.108}$$

The vectorizing of a skew symmetric matrix $S(\boldsymbol{a})$ yields

$$\mathrm{vec}S(\boldsymbol{a}) = \mathrm{vec}\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ a_3 \\ -a_2 \\ -a_3 \\ 0 \\ a_1 \\ a_2 \\ -a_1 \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}^{\mathsf{T}} \\ +\boldsymbol{e}_3^{\mathsf{T}} \\ -\boldsymbol{e}_2^{\mathsf{T}} \\ -\boldsymbol{e}_3^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} \\ +\boldsymbol{e}_1^{\mathsf{T}} \\ +\boldsymbol{e}_2^{\mathsf{T}} \\ -\boldsymbol{e}_1^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} \end{bmatrix}}_{J_S} \boldsymbol{a} \tag{29.109}$$

Therefore, we have

$$\mathrm{d}\overline{\mathbf{e}} = \mathrm{vec}\left(\mathrm{d}\mathsf{E}^{\mathsf{T}}\right) = -(I_3 \otimes R)\,J_S\,J_r\mathrm{d}\boldsymbol{b}_r + (\mathsf{E}\otimes I_3)\,J_S\,\mathrm{d}\boldsymbol{r} \tag{29.110}$$

Finally, we therefore find

$$\mathrm{d}\overline{\mathbf{e}} = \frac{\partial\overline{\mathbf{e}}}{\partial\boldsymbol{\theta}}\mathrm{d}\boldsymbol{\theta}\,, \tag{29.111}$$

with the Jacobian (**check signs**)

$$\boxed{\frac{\partial\overline{\mathbf{e}}}{\partial\boldsymbol{\theta}} = [-(I_3\otimes R)\,J_S\,J_r \mid (\mathsf{E}\otimes I_3)\,J_S]\,.} \tag{29.112}$$

29.4.4.3 The Jacobians of the constraints c w.r.t. observations y

With approximate values $\boldsymbol{x} = \boldsymbol{x}_0$ have from (29.15)

$$\mathrm{d}\boldsymbol{z} = \mathrm{d}\boldsymbol{b} = \mathrm{d}\begin{bmatrix} y_5 \\ y_6 \end{bmatrix} \quad \text{and} \quad \mathrm{d}A = \mathrm{d}\begin{bmatrix} y_1 & y_3 \\ y_2 & y_4 \end{bmatrix}\,. \tag{29.113}$$

Hence, from (29.62)

$$\boldsymbol{c} = X^{\mathsf{T}}\mathsf{E}\mathbf{z}(\boldsymbol{y}) + Z^{\mathsf{T}}(\boldsymbol{y})\mathsf{E}^{\mathsf{T}}\mathbf{x}_0 = \mathbf{0}\,, \tag{29.114}$$

we obtain

$$\mathrm{d}\boldsymbol{c} = X^{\mathsf{T}}\,\mathsf{E}(\boldsymbol{\theta})\,\mathrm{d}\begin{bmatrix} y_5 \\ y_6 \\ 0 \end{bmatrix} + \mathrm{d}\begin{bmatrix} y_1 & y_3 & y_5 \\ y_2 & y_4 & y_6 \\ 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}\,\mathsf{E}^{\mathsf{T}}(\boldsymbol{\theta})\,\mathbf{x}_0 \tag{29.115}$$

With the matrix

$$Y_1^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = [0,0,1] \otimes \mathsf{P}_0 \tag{29.116}$$

hence

$$\mathrm{d}\boldsymbol{b} = Y_1 \mathrm{d}\boldsymbol{y} \quad \text{or} \quad \mathrm{d}\begin{bmatrix} y_5 \\ y_6 \\ 0 \end{bmatrix} = Y_1 \mathrm{d}\boldsymbol{y} \tag{29.117}$$

the first summand can be written as

$$X^{\mathsf{T}}\,\mathsf{E}(\boldsymbol{\theta})\,\mathrm{d}\begin{bmatrix} y_5 \\ y_6 \\ 0 \end{bmatrix} = X^{\mathsf{T}}\,\mathsf{E}(\boldsymbol{\theta})\,Y_1^{\mathsf{T}}\,\mathrm{d}\boldsymbol{y}\,. \tag{29.118}$$

With the epipolar line $\boldsymbol{\ell}$ of $\mathbf{x}_0$ in the second image

$$\mathbf{l} = \mathsf{E}^{\mathsf{T}}\mathbf{x}_0 = \begin{bmatrix} \boldsymbol{l}_h \\ l_0 \end{bmatrix} \tag{29.119}$$

the second summand in (29.115) is

$$\mathrm{d}\begin{bmatrix} y_1 & y_2 & 0 \\ y_3 & y_4 & 0 \\ y_5 & y_6 & 0 \end{bmatrix}\mathbf{l} = \mathrm{d}\begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \\ y_5 & y_6 \end{bmatrix}\boldsymbol{l}_h = \begin{bmatrix} \mathrm{d}y_1 l_1 + \mathrm{d}y_2 l_2 \\ \mathrm{d}y_3 l_1 + \mathrm{d}y_4 l_2 \\ \mathrm{d}y_5 l_1 + \mathrm{d}y_6 l_2 \end{bmatrix} = Y_2^{\mathsf{T}}\mathrm{d}\boldsymbol{y} \tag{29.120}$$

with the matrix

$$Y_2^{\mathsf{T}} = \begin{bmatrix} l_1 & l_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & l_1 & l_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & l_1 & l_2 \end{bmatrix} = I_3 \otimes \boldsymbol{l}_h^{\mathsf{T}} \tag{29.121}$$

Hence, the Jacobian w.r.t. observations is

$$\boxed{B^{\mathsf{T}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = X^{\mathsf{T}}\,\mathsf{E}(\boldsymbol{\theta})\,Y_1^{\mathsf{T}} + Y_2^{\mathsf{T}}\,.} \tag{29.122}$$

We can simplify this, since the matrices Y_i have a special structure and are sparse. We first have:

$$B^{\mathsf{T}} = \begin{bmatrix} \mathsf{P}_0^{\mathsf{T}}\mathsf{E}(\boldsymbol{\theta}) \\ \mathbf{l}^{\mathsf{T}} \end{bmatrix}\,\mathsf{E}(\boldsymbol{\theta})\,([0,0,1] \otimes \mathsf{P}_0) + I_3 \otimes \boldsymbol{l}_h^{\mathsf{T}}\,. \tag{29.123}$$

This can be rearranged to

$$B^{\mathsf{T}} = [0,0,1] \otimes \begin{bmatrix} \mathsf{P}_0^{\mathsf{T}}\mathsf{E}(\boldsymbol{\theta})\mathsf{P}_0 \\ \mathbf{x}_0^{\mathsf{T}}\mathsf{E}(\boldsymbol{\theta})\mathsf{P}_0 \end{bmatrix} + I_3 \otimes \boldsymbol{l}_h^{\mathsf{T}} \tag{29.124}$$

This finally leads to

$$\boxed{\underset{3\times 6}{B^{\mathsf{T}}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = \begin{bmatrix} I_2 \otimes \boldsymbol{l}_h^{\mathsf{T}} & \mathsf{E}(\boldsymbol{\theta})_{1:2,1:2} \\ \mathbf{0}_4^{\mathsf{T}} & 2\boldsymbol{l}_h^{\mathsf{T}} \end{bmatrix}\,.} \tag{29.125}$$

29.4.4.4 Alternative derivation of the Jacobians of the constraints c w.r.t. observations y

With approximate values $\boldsymbol{x} = \boldsymbol{x}_0$ have from (29.14)

$$\mathrm{d}\mathbf{z}(\boldsymbol{y}) = \mathrm{d}\mathsf{A}\Delta\mathbf{x} = (\Delta\mathbf{x}^\mathsf{T} \otimes I_3)\mathrm{d}(\mathrm{vec}(\mathsf{A})) \quad \text{and} \quad \mathrm{d}A = \mathrm{d}\begin{bmatrix} y_1 & y_3 \\ y_2 & y_4 \end{bmatrix}. \tag{29.126}$$

Hence, from (29.62)

$$\boldsymbol{c} = X^\mathsf{T}\mathsf{E}\mathbf{z}(\boldsymbol{y}) + Z^\mathsf{T}(\boldsymbol{y})\mathsf{E}^\mathsf{T}\mathbf{x}_0 = \mathbf{0}\,, \tag{29.127}$$

we obtain

$$\mathrm{d}\boldsymbol{c} = X^\mathsf{T}\,\mathsf{E}(\boldsymbol{\theta})\,(\Delta\mathbf{x}^\mathsf{T} \otimes I_3)\mathrm{d}(\mathrm{vec}(\mathsf{A})) + \mathrm{d}\begin{bmatrix} y_1 & y_3 & y_5 \\ y_2 & y_4 & y_6 \\ 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \mathsf{E}^\mathsf{T}(\boldsymbol{\theta})\,\mathbf{x}_0 \tag{29.128}$$

The first summand can be rewritten the following way: With the matrix

$$Y_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = I_3 \otimes \mathsf{P}_0 \tag{29.129}$$

hence

$$\mathrm{d}(\mathrm{vec}(\mathsf{A})) = Y_1\mathrm{d}\boldsymbol{y} \tag{29.130}$$

the first summand can be written as

$$X^\mathsf{T}\,\mathsf{E}(\boldsymbol{\theta})\,(\Delta\mathbf{x}^\mathsf{T} \otimes I_3)(I_3 \otimes \mathsf{P}_0)\mathrm{d}\boldsymbol{y} = X^\mathsf{T}\,\mathsf{E}(\boldsymbol{\theta})\,(\Delta\mathbf{x}^\mathsf{T} \otimes \mathsf{P}_0)\mathrm{d}\boldsymbol{y}\,. \tag{29.131}$$

With the epipolar line $\boldsymbol{l}$ of $\mathbf{x}_0$ in the second image

$$\mathbf{l} = \mathsf{E}^\mathsf{T}\mathbf{x}_0 = \begin{bmatrix} \boldsymbol{l}_h \\ l_0 \end{bmatrix} \tag{29.132}$$

the second summand in (29.128) is

$$\mathrm{d}\begin{bmatrix} y_1 & y_2 & 0 \\ y_3 & y_4 & 0 \\ y_5 & y_6 & 0 \end{bmatrix}\mathbf{l} = \mathrm{d}\begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \\ y_5 & y_6 \end{bmatrix}\boldsymbol{l}_h = \begin{bmatrix} \mathrm{d}y_1 l_1 + \mathrm{d}y_2 l_2 \\ \mathrm{d}y_3 l_1 + \mathrm{d}y_4 l_2 \\ \mathrm{d}y_5 l_1 + \mathrm{d}y_6 l_2 \end{bmatrix} = Y_2^\mathsf{T}\mathrm{d}\boldsymbol{y} \tag{29.133}$$

with the matrix

$$Y_2^\mathsf{T} = \begin{bmatrix} l_1 & l_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & l_1 & l_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & l_1 & l_2 \end{bmatrix} = I_3 \otimes \boldsymbol{l}_h^\mathsf{T} \tag{29.134}$$

Hence, the Jacobian w.r.t. the observations is

$$\boxed{B^\mathsf{T} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = X^\mathsf{T}\,\mathsf{E}(\boldsymbol{\theta})\,Y_1^\mathsf{T} + Y_2^\mathsf{T}\,.} \tag{29.135}$$

We can simplify this, since the matrices Y_i have a special structure and are sparse. We first have:

$$B^\mathsf{T} = \begin{bmatrix} \mathsf{P}_0^\mathsf{T}\mathsf{E}(\boldsymbol{\theta}) \\ \mathbf{l}^\mathsf{T} \end{bmatrix}([0,0,1] \otimes \mathsf{P}_0) + I_3 \otimes \boldsymbol{l}_h^\mathsf{T}\,. \tag{29.136}$$

This can be rearranged to

$$B^\mathsf{T} = [0,0,1] \otimes \begin{bmatrix} \mathsf{P}_0^\mathsf{T}\mathsf{E}(\boldsymbol{\theta})\mathsf{P}_0 \\ \mathbf{x}_0^\mathsf{T}\mathsf{E}(\boldsymbol{\theta})\mathsf{P}_0 \end{bmatrix} + I_3 \otimes \boldsymbol{l}_h^\mathsf{T} \tag{29.137}$$

This finally leads to

$$\underset{3\times 6}{B^\mathsf{T}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = \begin{bmatrix} I_2 \otimes \boldsymbol{l}_h^\mathsf{T} & \mathsf{E}(\boldsymbol{\theta})_{1:2,1:2} \\ \mathbf{0}_4^\mathsf{T} & 2\boldsymbol{l}_h^\mathsf{T} \end{bmatrix}. \tag{29.138}$$

29.4.4.5 The Jacobians of the constraints $\boldsymbol{c}$ w.r.t. unknown parameters $\boldsymbol{\theta}$

We represent the uncertain rotation matrix via

$$\underline{R} = \exp(S(\underline{\boldsymbol{r}}))\mu_R \approx \mu_R + S(\mathrm{d}\boldsymbol{r})\mu_R\,. \tag{29.139}$$

Similarly we represent the uncertain basis vector $\mathbf{b}$ via

$$\underline{\mathbf{b}} = \mathsf{N}(\boldsymbol{\mu}_b + \underset{3\times 2}{J_r(\boldsymbol{\mu}_b)}\underline{\boldsymbol{b}}_r) \quad \text{with} \quad \underset{3\times 2}{J_r(\boldsymbol{\mu}_b)} = \mathsf{null}(\boldsymbol{\mu}_b^\mathsf{T})\,. \tag{29.140}$$

We therefore have

$$\mathrm{d}\mathbf{b} = J_r\,\mathrm{d}\boldsymbol{b}_r \quad \text{and} \quad \mathrm{d}R = S(\mathrm{d}\boldsymbol{r})R\,, \tag{29.141}$$

since $||\mathbf{b}|| = 1$ and assuming $R = \mu_R$. We collect the uncertain five parameters in the vector

$$\underline{\boldsymbol{\theta}} = \begin{bmatrix} \underline{\boldsymbol{b}}_r \\ \underline{\boldsymbol{r}} \end{bmatrix} \quad \text{with} \quad \mathbb{D}(\underline{\boldsymbol{\theta}}) = \Sigma_{\theta\theta} \tag{29.142}$$

Hence we have the differential of $\boldsymbol{c}$ w.r.t the unknown parameters

$$\mathrm{d}\boldsymbol{c} = X^\mathsf{T}\,\mathrm{d}\mathsf{E}(\boldsymbol{\theta})\,\mathbf{z} + Z^\mathsf{T}\,\mathrm{d}\mathsf{E}^\mathsf{T}(\boldsymbol{\theta})\,\mathbf{x}_0 \tag{29.143}$$

First we determine the differential of $\mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T} = R^\mathsf{T}\mathsf{S}(R\mathbf{b})$:

$$\mathrm{d}\mathsf{E}(\boldsymbol{\theta}) = \mathrm{d}\mathsf{S}(\mathbf{b})R^\mathsf{T} + \mathsf{S}(\mathbf{b})(S(\mathrm{d}\boldsymbol{r})R)^\mathsf{T} = \mathsf{S}(J_r\mathrm{d}\boldsymbol{b}_r)R^\mathsf{T} + \mathsf{E}S^\mathsf{T}(\mathrm{d}\boldsymbol{r})\,. \tag{29.144}$$

Therefore, the differential of $\boldsymbol{c}$ contains 4 summands

$$\mathrm{d}\boldsymbol{c} = \underbrace{X^\mathsf{T}\mathsf{S}(J_r\mathrm{d}\boldsymbol{b}_r)R^\mathsf{T}\mathbf{z}}_{K_1} + \underbrace{X^\mathsf{T}\mathsf{E}S^\mathsf{T}(\mathrm{d}\boldsymbol{r})\mathbf{z}}_{K_2} + \underbrace{Z^\mathsf{T}(\mathsf{S}(J_r\mathrm{d}\boldsymbol{b}_r)R^\mathsf{T})^\mathsf{T}\mathbf{x}_0}_{K_3} + \underbrace{Z^\mathsf{T}(\mathsf{E}S^\mathsf{T}(\mathrm{d}\boldsymbol{r}))^\mathsf{T}\mathbf{x}_0}_{K_4}\,. \tag{29.145}$$

Hence the Jacobian w.r.t. rotation parameters $\boldsymbol{\theta}$ is

$$D^\mathsf{T} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\theta}} = \left[\left(X^\mathsf{T}S^\mathsf{T}(R^\mathsf{T}\mathbf{z}) + Z^\mathsf{T}RS(\mathbf{x}_0)\right)J_r \,\middle|\, X^\mathsf{T}\mathsf{E}S(\boldsymbol{z}) + Z^\mathsf{T}S^\mathsf{T}(\mathsf{E}^\mathsf{T}\mathbf{x}_0)\right]. \tag{29.146}$$

Remark: We may expand the expressions an use the transformed point, the (non-normalized) normal of the epipolar plane and the relation between rotations and skew matrices:

$${}^1\mathbf{z} = R^\mathsf{T}\mathbf{z} \quad \text{and} \quad \mathbf{n}_x = \mathbf{x}_0 \times \mathbf{b} \quad \text{and} \quad R\mathsf{S}(\mathbf{d}) = \mathsf{S}(R\mathbf{d})R\,. \tag{29.147}$$

for simplifying the expressions

$$\begin{aligned}
K_1 &= X^{\mathsf{T}} S^{\mathsf{T}}({}^1\mathbf{z}) J_r \, \mathrm{d}\boldsymbol{b}_r = \begin{bmatrix} \mathsf{P}_0^{\mathsf{T}} \\ \mathbf{x}_0^{\mathsf{T}} \end{bmatrix} S^{\mathsf{T}}({}^1\mathbf{z}) J_r \, \mathrm{d}\boldsymbol{b}_r \\
K_2 &= X^{\mathsf{T}} \mathsf{E} S(\mathbf{z}) \, \mathrm{d}\boldsymbol{r} = \begin{bmatrix} \mathsf{P}_0^{\mathsf{T}} \\ \mathbf{x}_0^{\mathsf{T}} \end{bmatrix} \mathsf{S}(\mathbf{b}) R^{\mathsf{T}} S(\mathbf{z}) \, \mathrm{d}\boldsymbol{r} = \begin{bmatrix} \mathsf{P}_0^{\mathsf{T}} \mathsf{S}(\mathbf{b}) \\ \mathbf{n}_x^{\mathsf{T}} \end{bmatrix} \mathsf{S}({}^1\mathbf{z}) R^{\mathsf{T}} \, \mathrm{d}\boldsymbol{r} \\
K_3 &= Z^{\mathsf{T}} R S(\mathbf{x}_0) J_r \, \mathrm{d}\boldsymbol{b}_r = \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} \\ \mathbf{z}^{\mathsf{T}} \end{bmatrix} R S(\mathbf{x}_0) J_r \, \mathrm{d}\boldsymbol{b}_r = \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} R \\ {}^1\mathbf{z}^{\mathsf{T}} \end{bmatrix} S(\mathbf{x}_0) J_r \, \mathrm{d}\boldsymbol{b}_r \\
K_4 &= Z^{\mathsf{T}} \mathsf{S}(\mathrm{d}\boldsymbol{r}) \mathsf{E}^{\mathsf{T}} \mathbf{x}_0 && (29.148) \\
&= Z^{\mathsf{T}} \mathsf{S}^{\mathsf{T}}(\mathsf{E}^{\mathsf{T}} \mathbf{x}_0) \mathrm{d}\boldsymbol{r} && (29.149) \\
&= Z^{\mathsf{T}} R R^{\mathsf{T}} \mathsf{S}^{\mathsf{T}}(R \mathsf{S}^{\mathsf{T}}(\mathbf{b}) \mathbf{x}_0) \mathrm{d}\boldsymbol{r} && (29.150) \\
&= (Z^{\mathsf{T}} R) \, R^{\mathsf{T}} \mathsf{S}(R \mathsf{S}(\mathbf{b}) \mathbf{x}_0) \mathrm{d}\boldsymbol{r} && (29.151) \\
&= (Z^{\mathsf{T}} R) \, R^{\mathsf{T}} \mathsf{S}^{\mathsf{T}}(R \mathbf{n}_x) \mathrm{d}\boldsymbol{r} && (29.152) \\
&= \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} R \\ \mathbf{z}^{\mathsf{T}} R \end{bmatrix} R^{\mathsf{T}} \mathsf{S}^{\mathsf{T}}(R \mathbf{n}_x) \mathrm{d}\boldsymbol{r} && (29.153) \\
&= \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} R \\ {}^1\mathbf{z}^{\mathsf{T}} \end{bmatrix} \mathsf{S}^{\mathsf{T}}(\mathbf{n}_x) R^{\mathsf{T}} \, \mathrm{d}\boldsymbol{r}
\end{aligned}$$

or

$$D^{\mathsf{T}} = \left[\left(\begin{bmatrix} \mathsf{P}_0^{\mathsf{T}} S^{\mathsf{T}}({}^1\mathbf{z}) \\ ({}^1\mathbf{z} \times \mathbf{x}_0)^{\mathsf{T}} \end{bmatrix} + \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} R S(\mathbf{x}_0) \\ ({}^1\mathbf{z} \times \mathbf{x}_0)^{\mathsf{T}} \end{bmatrix} \right) J_r \middle| \left(\begin{bmatrix} \mathsf{P}_0^{\mathsf{T}} \mathsf{S}(\mathbf{b}) \mathsf{S}({}^1\mathbf{z}) \\ (\mathbf{n}_x \times {}^1\mathbf{z})^{\mathsf{T}} \end{bmatrix} + \begin{bmatrix} \mathsf{A}_0^{\mathsf{T}} R \mathsf{S}^{\mathsf{T}}(\mathbf{n}_x) \\ (\mathbf{n}_x \times {}^1\mathbf{z})^{\mathsf{T}} \end{bmatrix} \right) R^{\mathsf{T}} \right] \tag{29.154}$$

However, these expressions are computationally more demanding. ◇

29.4.4.6 The Jacobian $\partial(\boldsymbol{C}_r^{\mathsf{T}})/\partial \boldsymbol{C}^{\mathsf{T}}$

The Jacobian of the selected lines are the selected Jacobians of all lines.

The constraint matrix with selected rows follows from

$$\underset{5\times 9}{C_r^{\mathsf{T}}} = \underset{5\times 6}{G_r^{\mathsf{T}}} \, \underset{6\times 9}{C^{\mathsf{T}}} . \tag{29.155}$$

where the selection matrix consists of 5 unit vectors of length 6:

$$G_r^{\mathsf{T}} = \begin{bmatrix} \boldsymbol{e}_{r_1}^{\mathsf{T}} \\ \boldsymbol{e}_{r_2}^{\mathsf{T}} \\ \boldsymbol{e}_{r_3}^{\mathsf{T}} \\ \boldsymbol{e}_{r_4}^{\mathsf{T}} \\ \boldsymbol{e}_{r_5}^{\mathsf{T}} \end{bmatrix} := \begin{bmatrix} \boldsymbol{e}_{r_1}^{[6],\mathsf{T}} \\ \boldsymbol{e}_{r_2}^{[6],\mathsf{T}} \\ \boldsymbol{e}_{r_3}^{[6],\mathsf{T}} \\ \boldsymbol{e}_{r_4}^{[6],\mathsf{T}} \\ \boldsymbol{e}_{r_5}^{[6],\mathsf{T}} \end{bmatrix} \quad \text{with} \quad r_k \in \{1, \ldots, 6\} \quad \text{and} \quad G_r^{\mathsf{T}} G_r = I_5 . \tag{29.156}$$

We again determine the Jacobians for each row separately. For the r_k-th constraint we have

$$\boldsymbol{c}_{r_k}^{\mathsf{T}} = \boldsymbol{e}_{r_k}^{\mathsf{T}} C^{\mathsf{T}} \quad \text{or} \quad \boldsymbol{c}_{r_k} = C \boldsymbol{e}_{r_k} . \tag{29.157}$$

or

$$\boldsymbol{c}_{rk} = (\boldsymbol{e}_{r_k}^{\mathsf{T}} \otimes I_9) \, \mathrm{vec} C . \tag{29.158}$$

Therefore we obtain the Jacobian

$$\frac{\partial \boldsymbol{c}_{rk}}{\partial \mathrm{vec} C} = \boldsymbol{e}_{r_k}^{\mathsf{T}} \otimes I_9 . \tag{29.159}$$

29.4.5 The algorithm

The algorithm is given below.

Algorithm 6: Covariance matrix of $(\mathbf{b}, R)$ from two ACs;
Σ=`CovM_bR_from_2_ACs`$(U_0, V_0, \mathsf{A}, \Sigma_{aa}, \mathbf{b}, R, \boldsymbol{s})$

Input: Reference points, 2×2 matrices: $U_0 = \begin{bmatrix} \boldsymbol{x}_{01}^\mathsf{T} \\ \boldsymbol{x}_{02}^\mathsf{T} \end{bmatrix}$, $V_0 = \begin{bmatrix} \boldsymbol{z}_{01}^\mathsf{T} \\ \boldsymbol{z}_{02}^\mathsf{T} \end{bmatrix}$;
Affinity tensor $3 \times 3 \times 2$ with homogeneous matrices: $\mathsf{A}_i = \mathsf{A}(:,:,i), i = 1, 2$;
Covariance matrix Σ_{aa} of uncertain affine correspondences;
Relative orientation $(\mathbf{b}, R)$;
Vector $\boldsymbol{s}$ of indices for selection of constraints.
Condition: The constraint matrix has full rank.
Output: 5×5 Covariance matrix $\Sigma = \mathbb{D}(\boldsymbol{p})$, with $\boldsymbol{p} = [\boldsymbol{b}_r; \boldsymbol{r}]$.

Essential matrix: $\mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T}$;
Fitted points: $V = V_0 + \mathsf{A}(1:2, 3, :)$, see (29.16);
Homogeneous coordinates: $\mathsf{U}_0 = [U_0, \mathbf{1}]$, $\mathsf{V} = [V, \mathbf{1}]$;
Ancillary vectors: $\mathsf{L} = \mathsf{U}_0\mathsf{E} = \begin{bmatrix} \mathbf{l}_1^\mathsf{T} \\ \mathbf{l}_2^\mathsf{T} \end{bmatrix}$, see (29.132);
Ancillary matrix: $\underset{3\times2}{\mathsf{P}_0} = [I_2, \mathbf{0}]^\mathsf{T}$, see (29.61) ;
for $i = 1, 2$ **do**
  Jacobian: $\partial(\boldsymbol{c}_i)/\partial\boldsymbol{y}_i = B_i^\mathsf{T} = [\mathsf{P}_0, \mathbf{x}_i]^\mathsf{T}\, \mathsf{E}\, (\boldsymbol{e}_3^\mathsf{T} \otimes P_0 + I_3 \otimes \boldsymbol{l}_{hi})$, see (29.122)
end
Jacobian: $\partial(\boldsymbol{c})/\partial\boldsymbol{y} = B^\mathsf{T} = \mathrm{Diag}(\{B_i^\mathsf{T}\})$;
Jacobian: $J_r = \mathsf{null}(\mathbf{b}^\mathsf{T})$;
for $i = 1, 2$ **do**
  Ancillary matrix: $A_{0i} = \begin{bmatrix} \mathsf{A}(1:2, 1:2, i) \\ \mathbf{0}^\mathsf{T} \end{bmatrix}$, see (29.61);
  Ancillary matrices $X_i = [\mathsf{P}_0, \mathbf{x}_i]$, $Z_i = [A_{0i}, \mathbf{z}_i]$, see (29.61);
  Ancillary matrices $K_1 = X_i^\mathsf{T} S^\mathsf{T}(R\mathbf{z}_i)$, $K_2 = Z_i^\mathsf{T} R\mathsf{S}(\mathbf{x}_0)$, see (29.146);
  Ancillary matrices $K_3 = X_i^\mathsf{T} \mathsf{E} S^\mathsf{T}(\mathbf{z}_i)$, $K_4 = Z_i^\mathsf{T} \mathsf{S}^\mathsf{T}(\mathsf{E}^\mathsf{T}\mathbf{x}_0)$, see (29.146);
  Jacobians (3×5): $\partial(\boldsymbol{c}_i)/\partial\boldsymbol{\theta} = D_i^\mathsf{T} = [(K_1 + K_2)J_r, K_3 + K_4]$, see (29.146);
end
Jacobian: $\underset{6\times5}{D} = \begin{bmatrix} D_1^\mathsf{T} \\ D_2^\mathsf{T} \end{bmatrix}$;
Selection matrix: $G_r^\mathsf{T} = G_r^\mathsf{T}(\boldsymbol{s})$;
(Inverse) Jacobians of selected constraints: $H = (G_r^\mathsf{T} D)^{-1}$, $B_r^\mathsf{T} = G_r^\mathsf{T} B^\mathsf{T}$;
Covariance matrix (5×5): $\Sigma = HB_r^\mathsf{T}\Sigma_{aa}B_r H^\mathsf{T}$, see (29.82);

29.5 Precision of relative orientation with focal length from pairs of affine matches

29.5.1 The constraint – version 1

We use the same constraint as for the case of the calibrated camera. However, we

1. use the image coordinates

$$\boldsymbol{x} := {}^{i}\boldsymbol{x}' \quad \text{and} \quad \boldsymbol{z} := {}^{i}\boldsymbol{x}'' \tag{29.160}$$

or

$$\mathbf{x} := {}^{i}\mathbf{x}' = \begin{bmatrix} {}^{i}x' \\ {}^{i}y' \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{z} := {}^{i}\mathbf{x}'' = \begin{bmatrix} {}^{i}x'' \\ {}^{i}y'' \\ 1 \end{bmatrix} \tag{29.161}$$

together with the affinity referring to the image coordinates, and

2. use the fundamental matrix

$$\mathsf{F}(\mathbf{b}, R, \phi) = \mathsf{Q}\mathsf{E}\mathsf{Q} \quad \text{with} \quad \mathsf{Q} = \mathrm{Diag}([1, 1, f]) \quad \text{and} \quad \mathsf{E} = \mathsf{S}(\mathbf{b})R^{\mathsf{T}} . \tag{29.162}$$

3. We use all constraints, thus do not reduce them to 5.

Thus we have the point constraint

$$\mathbf{x}^{\mathsf{T}}\mathsf{F}(\mathbf{b}, R, f)\mathbf{z} = (\mathbf{x}^{\mathsf{T}} \otimes \mathbf{z}^{\mathsf{T}})\bar{\mathbf{f}} = 0 \tag{29.163}$$

and the combined constraint

$$\boldsymbol{c} = X^{\mathsf{T}}\mathsf{F}\mathbf{z} + Z^{\mathsf{T}}\mathsf{F}^{\mathsf{T}}\mathbf{x} = \mathbf{0} . \tag{29.164}$$

or

$$\overline{C}^{\mathsf{T}}\,\bar{\mathbf{f}} = 0 , \tag{29.165}$$

with

$$\overline{C}^{\mathsf{T}} = X^{\mathsf{T}} \otimes \mathbf{z}^{\mathsf{T}} + \mathbf{x}^{\mathsf{T}} \otimes Z^{\mathsf{T}} . \tag{29.166}$$

Therefore, the Jacobians of the constraints can be taken from the case without focal length, except for

1. Using the 6×9 matrix $\overline{C}^{\mathsf{T}}$ instead of the 5×9 constraint matrix.
2. Using the fundamental matrix F instead of the essential matrix, hence the epipolar line

$$\mathbf{l} = \mathsf{F}^{\mathsf{T}}\mathbf{x}_0 \quad \text{and} \quad \boldsymbol{l}_h = \mathbf{l}_{1:2} . \tag{29.167}$$

Therefore, the Jacobian B is

$$\mathsf{B}^{\mathsf{T}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = \begin{bmatrix} I_2 \otimes \boldsymbol{l}_h^{\mathsf{T}} & \mathsf{F}(\boldsymbol{\theta})_{1:2,1:2} \\ \mathbf{0}_4^{\mathsf{T}} & 2\boldsymbol{l}_h^{\mathsf{T}} \end{bmatrix} . \tag{29.168}$$

3. Extending the Jacobian w.r.t. unknowns for the 6-th parameter f, see below.

29.5.2 The constraint – Version 2

29.5.2.1 The constraints

We use the same constraint as for the case of the calibrated camera. However, we

1. use the image coordinates

$$\boldsymbol{x} := {}^{i}\boldsymbol{x}' \quad \text{and} \quad \boldsymbol{z} := {}^{i}\boldsymbol{x}'' \tag{29.169}$$

or

$$\mathbf{x} := \mathbf{x}(f) = {}^{c}\mathbf{x}' = \begin{bmatrix} x' \\ y' \\ f \end{bmatrix} \quad \text{and} \quad \mathbf{z} := \mathbf{z}(f) = {}^{c}\mathbf{x}'' = \begin{bmatrix} x'' \\ y'' \\ f \end{bmatrix} \tag{29.170}$$

together with the affinity referring to the image coordinates,

2. We use all constraints, thus do not reduce them to 5.

Thus we have the point constraint

$$\mathbf{x}^{\mathsf{T}}(f)\, \mathsf{E}(\mathbf{b}, R)\, \mathbf{z}(f) = (\mathbf{x}^{\mathsf{T}}(f) \otimes \mathbf{z}^{\mathsf{T}}(f))\, \overline{\mathbf{e}} = 0 \tag{29.171}$$

and the combined constraint

$$\boldsymbol{c} = X^{\mathsf{T}}(f)\, \mathsf{E}\, \mathbf{z}(f) + Z^{\mathsf{T}}(f)\, \mathsf{E}^{\mathsf{T}}\, \mathbf{x}(f) = \mathbf{0}\,. \tag{29.172}$$

with

$$X^{\mathsf{T}} = \begin{bmatrix} I_2 & \boldsymbol{x} \\ \mathbf{0}^{\mathsf{T}} & f \end{bmatrix} \quad \text{and} \quad Z^{\mathsf{T}} = \begin{bmatrix} A & \boldsymbol{z} \\ \mathbf{0}^{\mathsf{T}} & f \end{bmatrix} \tag{29.173}$$

Hence we have

$$C^{\mathsf{T}}(f)\, \overline{\mathbf{e}} = \mathbf{0} \quad \text{or} \quad \overline{C}^{\mathsf{T}}(f)\, \overline{\mathbf{e}} = \mathbf{0}\,, \tag{29.174}$$

with

$$C^{\mathsf{T}} = \mathbf{z}^{\mathsf{T}} \otimes X^{\mathsf{T}} + Z^{\mathsf{T}} \otimes \mathbf{x}^{\mathsf{T}}\,. \tag{29.175}$$

or

$$\overline{C}^{\mathsf{T}} = X^{\mathsf{T}} \otimes \mathbf{z}^{\mathsf{T}} + \mathbf{x}^{\mathsf{T}} \otimes Z^{\mathsf{T}}\,. \tag{29.176}$$

Therefore, the Jacobians of the constraints can be taken from the case without focal length, except for

1. Using the 6×9 constraint matrix instead of the 5×9 constraint matrix.
2. Using the essential matrix E, hence the epipolar line

$$\mathbf{l} = \mathsf{E}^{\mathsf{T}}\mathbf{x}_0 \quad \text{and} \quad \boldsymbol{l}_h = \mathbf{l}_{1:2}\,. \tag{29.177}$$

Therefore, the Jacobian B is

$$\mathsf{B}^{\mathsf{T}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}} = \begin{bmatrix} I_2 \otimes \boldsymbol{l}_h^{\mathsf{T}} & \mathsf{E}(\boldsymbol{\theta})_{1:2,1:2} \\ \mathbf{0}_4^{\mathsf{T}} & 2\boldsymbol{l}_h^{\mathsf{T}} \end{bmatrix}. \tag{29.178}$$

3. Extending the Jacobian w.r.t. unknowns for the 6-th parameter f, see below.

29.5.2.2 The Jacobians

The 6 parameters in $\mathrm{d}\boldsymbol{\theta}$ now consist of the 5 parameters $\mathrm{d}\boldsymbol{\theta}_1$ and the focal length $\mathrm{d}\theta = \mathrm{d}f$:

$$\mathrm{d}\boldsymbol{\theta} = \begin{bmatrix} \mathrm{d}\boldsymbol{\theta}_1 \\ \mathrm{d}\theta_2 \end{bmatrix} = \begin{bmatrix} \mathrm{d}\boldsymbol{b}_r \\ \mathrm{d}\boldsymbol{r} \\ \mathrm{d}f \end{bmatrix} \tag{29.179}$$

Hence taking the differential of

$$\boldsymbol{c} = X^{\mathsf{T}}\mathsf{E}\mathbf{z} + Z^{\mathsf{T}}\mathsf{E}^{\mathsf{T}}\mathbf{x}_0 = \mathbf{0}\,. \tag{29.180}$$

uses (29.146)

$$D_1^{\mathsf{T}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\theta}_1} = \left[\left(X^{\mathsf{T}} S^{\mathsf{T}}(R^{\mathsf{T}}\mathbf{z}) + Z^{\mathsf{T}} R S(\mathbf{x}_0) \right) J_r \,\middle|\, X^{\mathsf{T}}\mathsf{E}S(\boldsymbol{z}) + Z^{\mathsf{T}} S^{\mathsf{T}}(\mathsf{E}^{\mathsf{T}}\mathbf{x}_0) \right] \tag{29.181}$$

and

$$D_2^\mathsf{T} = \frac{\partial \boldsymbol{c}}{\partial \theta_2} = \mathrm{Diag}(\boldsymbol{e}_3)\ \mathsf{E}\ \mathbf{z}_i + X_i^\mathsf{T}\ \mathsf{E}\ \boldsymbol{e}_3 + \mathrm{Diag}(\boldsymbol{e}_3)\ \mathsf{E}^\mathsf{T}\ \mathbf{x}_i + Z_i^\mathsf{T}\ \mathsf{E}^\mathsf{T}\ \boldsymbol{e}_3 \tag{29.182}$$

Therefore, the differential $\mathrm{d}\boldsymbol{c}$ now consists of 5 terms, the first four referring to $[\mathrm{d}\boldsymbol{b}_r, \mathrm{d}\boldsymbol{r}]$ the last referring to $\mathrm{d}f$

$$\mathrm{d}\boldsymbol{c} = \sum_j K_j \tag{29.183}$$

with

$$\begin{aligned}
K_1 &= X^\mathsf{T}\ \mathsf{S}^\mathsf{T}(R^\mathsf{T}\mathbf{z})\ J_r\ \mathrm{d}\boldsymbol{b}_r && (29.184)\\
K_2 &= X^\mathsf{T}\ \mathsf{E}\ S^\mathsf{T}(\mathbf{z})\ \mathrm{d}\boldsymbol{r} && (29.185)\\
K_3 &= Z^\mathsf{T}\ R\ \mathsf{S}(\mathbf{x}_0)\ J_r\ \mathrm{d}\boldsymbol{b}_r && (29.186)\\
K_4 &= Z^\mathsf{T}\ S^\mathsf{T}(\mathsf{E}^\mathsf{T}\ \mathbf{x}_0)\ \mathrm{d}\boldsymbol{r} && (29.187)\\
K_5 &= \mathrm{Diag}(\boldsymbol{e}_3)\ \mathsf{E}\ \mathbf{z}_i + X_i^\mathsf{T}\ \mathsf{E}\ \boldsymbol{e}_3 + \mathrm{Diag}(\boldsymbol{e}_3)\ \mathsf{E}^\mathsf{T}\ \mathbf{x}_i + Z_i^\mathsf{T}\ \mathsf{E}^\mathsf{T}\ \boldsymbol{e}_3 && (29.188)
\end{aligned}$$

as to be shown.

Hence, we observe that the matrices K_1 to K_4 algebraically are the same as before, taking into account that the coordinates in all cases refer to the camera system. However, here the camera coordinates are *not* Euclideanly normalized, hence:

$$\mathbf{x} := \begin{bmatrix} x' \\ y' \\ f \end{bmatrix} \quad \text{and} \quad \mathbf{z} := \begin{bmatrix} x'' \\ y'' \\ f \end{bmatrix}. \tag{29.189}$$

Therefore the Jacobians here numerically are larger by a factor f^2, than those Jacobians derived for the calibrated case.

For deriving K_5

$$K_5 = \frac{\partial \boldsymbol{c}}{\partial f} = \frac{\partial}{\partial f}(X^\mathsf{T}(f)\ \mathsf{E}\ \mathbf{z}(f) + Z^\mathsf{T}(f)\ \mathsf{E}^\mathsf{T}\ \mathbf{x}(f)) \tag{29.190}$$

we use the following simplifying relations, due to the Euclidean normalization of the coordinates

$$\begin{aligned}
\mathrm{d}\mathbf{x}(f) &= \boldsymbol{e}_3\ \mathrm{d}f && (29.191)\\
\mathrm{d}\mathbf{z}(f) &= \boldsymbol{e}_3\ \mathrm{d}f && (29.192)\\
\mathrm{d}X^\mathsf{T}(f) &= \mathrm{Diag}(\boldsymbol{e}_3)\ \mathrm{d}f && (29.193)\\
\mathrm{d}Z^\mathsf{T}(f) &= \mathrm{Diag}(\boldsymbol{e}_3)\ \mathrm{d}f && (29.194)
\end{aligned}$$

and obtain

$$K_5 = \mathrm{Diag}(\boldsymbol{e}_3)\mathsf{E}\ \mathbf{z} + X^\mathsf{T}\ \mathsf{E}\ \boldsymbol{e}_3 + \mathrm{Diag}(\boldsymbol{e}_3)\mathsf{E}^\mathsf{T}\ \mathbf{x}_0 + Z^\mathsf{T}\ \mathsf{E}^\mathsf{T}\ \boldsymbol{e}_3 \tag{29.195}$$

29.5.2.3 A short derivation for the Jacobian of the affine constraints w.r.t. focal length

Given are two affine correspondences (ACs):

$$\mathrm{AC}_i : \quad \{\boldsymbol{x}, \boldsymbol{z}, A, \Sigma_{aa}\}_i \tag{29.196}$$

with

- the coordinates $\boldsymbol{x}$ and $\boldsymbol{z}$ in the left and the right image, referring to the principal point in pixel coordinates
- the uncertain affinity
$$A = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} \tag{29.197}$$
and
- the covariance matrix Σ_{aa} of the six parameters $[a_1, ..., a_4, \boldsymbol{z}^\mathsf{T}]$. Hence, we treat $\boldsymbol{x}$ as fixed value.

The three constraints for the two ACs can be written as

$$\boldsymbol{c}_i = X_i^\mathsf{T} \, \mathsf{E} \, \mathbf{z}_i + Z_i^\mathsf{T} \, \mathsf{E} \, \mathbf{x}_i = \mathbf{0} \tag{29.198}$$

where

- The homogeneous coordinates of the directions in the camera system are
$$\mathbf{x}_i = \begin{bmatrix} \boldsymbol{x}_i \\ f \end{bmatrix} \quad \text{and} \quad \mathbf{z}_i = \begin{bmatrix} \boldsymbol{z}_i \\ f \end{bmatrix} \tag{29.199}$$
Care has to be taken for the sign of f. We have a right hand system in the image plane. Then the Z-axis of the camera points away from the scene. If the images are in taking position (the sky is at the top), then the image plane in the camera coordinate system is at $Z = f$, where $f < 0$.
- the matrices are
$$X_i = \begin{bmatrix} I_2 & \boldsymbol{x}_i \\ \mathbf{0}^\mathsf{T} & f \end{bmatrix} \quad \text{and} \quad Z_i = \begin{bmatrix} A & \boldsymbol{z}_i \\ \mathbf{0}^\mathsf{T} & f \end{bmatrix} \tag{29.200}$$

Observe, the epipolar constraint is in the third row, and taken with a factor 2.

The constraint for the ACs are

$$\boldsymbol{c}_i = X_i^\mathsf{T} \, \mathsf{E} \, \mathbf{z}_i + Z_i^\mathsf{T} \, \mathsf{E}^\mathsf{T} \, \mathbf{x}_i \,, \tag{29.201}$$

We have the derivatives w.r.t. f

$$\frac{\partial X_i^\mathsf{T}}{\partial f} = \mathrm{Diag}(\boldsymbol{e}_3) \tag{29.202}$$
$$\frac{\partial Z_i^\mathsf{T}}{\partial f} = \mathrm{Diag}(\boldsymbol{e}_3) \tag{29.203}$$
$$\frac{\partial \mathbf{x}_i}{\partial f} = \boldsymbol{e}_3 \tag{29.204}$$
$$\frac{\partial \mathbf{z}_i}{\partial f} = \boldsymbol{e}_3 \,. \tag{29.205}$$

Therefore, the differential is

$$\mathrm{d}\boldsymbol{c}_i = \mathrm{d}X_i^\mathsf{T} \, \mathsf{E} \, \mathbf{z}_i + X_i^\mathsf{T} \, \mathsf{E} \, \mathrm{d}\mathbf{z}_i + \mathrm{d}Z_i^\mathsf{T} \, \mathsf{E}^\mathsf{T} \, \mathbf{x}_i + Z_i^\mathsf{T} \, \mathsf{E}^\mathsf{T} \, \mathrm{d}\mathbf{x}_i \tag{29.206}$$
$$= \mathrm{Diag}(\boldsymbol{e}_3) \, \mathsf{E} \, \mathbf{z}_i + X_i^\mathsf{T} \, \mathsf{E} \, \boldsymbol{e}_3 + \mathrm{Diag}(\boldsymbol{e}_3) \, \mathsf{E}^\mathsf{T} \, \mathbf{x}_i + Z_i^\mathsf{T} \, \mathsf{E}^\mathsf{T} \, \boldsymbol{e}_3 \tag{29.207}$$

29.5.3 Generating ACs

We can easily generate an AC. For this we need to specify

1. the relative pose $(R, \boldsymbol{B})$ of the two images
2. the plane $\mathcal{A}(\mathbf{A})$
3. a reference point $\mathcal{X}_0(\mathbf{x}_0)$ in the first image
4. a small offset $\boldsymbol{b}$ for determining the other reference point $\mathcal{Z}_0(\boldsymbol{z}_0)$.

We preform the following calculations

1. Determine the intersection point of the plane $\mathcal{A}$ with the projection ray $\mathcal{L}_{x_0}$ of $\mathcal{X}_0$:

$$\mathcal{X} = \mathcal{A} \cap \mathcal{L}_{x_0} \tag{29.208}$$

   which is for $\mathbf{A}^\mathsf{T} = [\boldsymbol{N}^\mathsf{T}, -S]$

$$\mathbf{X} = \mathsf{\Pi}^\mathsf{T}(\mathbf{A})\overline{\mathsf{Q}}^\mathsf{T}\mathbf{x}_0 = \begin{bmatrix} -SI_3 & S^\mathsf{T}(\boldsymbol{N}) \\ \boldsymbol{N}^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -S\mathbf{x}_0 \\ \boldsymbol{N}^\mathsf{T}\mathbf{x}_0 \end{bmatrix} \tag{29.209}$$

2. determine the corresponding point $\boldsymbol{z}$ in the second image

$$\mathbf{z} = \mathsf{P}''\mathbf{X}\,. \tag{29.210}$$

3. determine the reference point $\boldsymbol{z}_0$

$$\boldsymbol{z}_0 = \boldsymbol{z} - \boldsymbol{b}\,. \tag{29.211}$$

4. determine the affinity A from (29.14), where the Jacobians are evaluated at $(\boldsymbol{x}_0, \boldsymbol{z})$:

$$A = J_2 J_1^{-1} \tag{29.212}$$

   with

$$J_1 = J_c(\mathbf{x}_0)\; J_Z \quad J_2 = J_c(\mathbf{z})\; R\; J_Z \quad \text{and} \quad J_Z = \begin{bmatrix} I_2 \\ \boldsymbol{n}^\mathsf{T} \end{bmatrix} \tag{29.213}$$

For choosing a plausible covariance matrix Σ_{aa} for the 6 parameters $(\mathsf{A}, \boldsymbol{b})$ we take one from a LSM result.

For generating noisy ACs, we sample 6-dimensional deviations $\underline{\Delta \boldsymbol{a}}$ from $\mathcal{N}(\mathbf{0}, \Sigma_{aa})$ to obtain samples

$$\underline{\mathsf{A}} = \begin{bmatrix} \underline{A} & \underline{\boldsymbol{b}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \quad \text{with} \quad \underline{A} = A + \underline{\Delta A} \quad \text{and} \quad \underline{\boldsymbol{b}} = \boldsymbol{b} + \underline{\Delta \boldsymbol{b}}\,. \tag{29.214}$$

29.6 Local affinities of homographies

Local affinities derived from least squares matching should be similar to the local affinity derived by differentiation, if projective component is not too large.

Let a homography be

$$\mathbf{y} = \mathsf{H}\mathbf{x}\,. \tag{29.215}$$

We want to derive the local affinity at some given pair $(\boldsymbol{x}_0, \boldsymbol{y}_0)$. For this we linearize and start with

$$\mathsf{H} = \begin{bmatrix} H_{11} & \boldsymbol{h}_{12} \\ \boldsymbol{h}_{21}^\mathsf{T} & h_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}. \tag{29.216}$$

Then we have

$$\boldsymbol{y} = \frac{H_{11}\boldsymbol{x} + \boldsymbol{h}_{12}}{\boldsymbol{h}_{21}^\mathsf{T}\boldsymbol{x} + h_{22}}\,. \tag{29.217}$$

For deriving the Jacobian $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ we use

$$\boldsymbol{y}(\boldsymbol{h}_{21}^{\mathsf{T}}\boldsymbol{x} + h_{22}) = \mathsf{H}_{11}\boldsymbol{x} + \boldsymbol{h}_{12}\,, \tag{29.218}$$

take the total differential:

$$\mathrm{d}\boldsymbol{y}(\boldsymbol{h}_{21}^{\mathsf{T}}\boldsymbol{x} + h_{22}) + \boldsymbol{y}(\boldsymbol{h}_{21}^{\mathsf{T}}\mathrm{d}\boldsymbol{x}) = \mathsf{H}_{11}\mathrm{d}\boldsymbol{x} \tag{29.219}$$

and solve for $\mathrm{d}\boldsymbol{y}$:

$$\mathrm{d}\boldsymbol{y} = \underbrace{\frac{\mathsf{H}_{11} - \boldsymbol{y}\boldsymbol{h}_{21}^{\mathsf{T}}}{\boldsymbol{h}_{21}^{\mathsf{T}}\boldsymbol{x} + h_{22}}}_{\mathsf{A}(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x} \tag{29.220}$$

Hence the local affinity at some pair $(\boldsymbol{x}_0, \boldsymbol{y}_0))$ is given by

$$\boldsymbol{y} - \boldsymbol{y}_0 = \mathsf{A}(\boldsymbol{x} - \boldsymbol{x}_0) \tag{29.221}$$

or

$$\boldsymbol{y} = \mathsf{A}\boldsymbol{x} + (\boldsymbol{y}_0 - \mathsf{A}\boldsymbol{x}_0)\,. \tag{29.222}$$

29.7 Experiments

29.7.1 Goals

The goal of the experiments is to demonstrate the validity and the usefulness of the approach. We therefore want to answer the following questions:

- Prove the validity of the derivation of the relative pose and its uncertainty from two ACs.
- Compare the usefulness of the derivation of the essential matrix from two ACs with that from five point correspondences (PCs)

29.7.2 Validity of the estimation of the essential matrix from two ACs

This is realized in the MATLAB script `demo_2ACs_2_E_precision.m`. It performs the following steps:

1. The first camera is set to $\mathsf{P} = [\mathsf{I}_3 \mid \mathbf{0}]$.
2. A second camera with $\mathsf{P} = [\mathsf{R} \mid -\boldsymbol{B}]$ is generated randomly.
3. Two planes $(\boldsymbol{X}_0, \boldsymbol{N})$ are generated randomly.
4. A small offset $\boldsymbol{b}$ for the affinity is generated randomly.
5. Two covariance matrices

   $$\Sigma_{aa} = \mathrm{Diag}([\sigma_a^2, \sigma_a^2, \sigma_a^2, \sigma_a^2, \sigma_b^2, \sigma_b^2]) \tag{29.223}$$

   with some $\sigma_{f'}$ representative for the image patch

   $$\sigma_a = \frac{\sqrt{12}}{\sigma_{f'} s^2} \quad \text{and} \quad \sigma_a = \frac{1}{\sigma_{f'} s} \tag{29.224}$$

   is generated depending on a prespecified scale s.
6. Two uncertain ACs are generated. The noise is varied for different samples within an experiment.

7. The essential matrix and the 5×5 covariance matrix for the basis direction and the rotational uncertainty are determined.
8. Using a sample of K the correctness of the implementation is checked for the covariance matrix and the bias.

For small noise of $\sigma_{f'} = 500$ [gr/pixel] and $K = 1000$ samples no deviation of the empirical from the theoretical covariance matrix and no bias was observed.

29.7.3 Compare the usefulness of deriving the essential Matrix from two ACs

This test refers to real data.

1. We use the multi-view data set of (https://icwww.epfl.ch/multiview/denseMVS.html), see Strecha et al. (2008).
2. We selected appropriate image pairs and performed a bundle adjustment to obtain matches between the images using AURELO, see Läbe and Förstner (2006). This yields Lowe-keypoints with position, scale and orientation.
3. We used 5-tupels of matches to verify the 5-point algorithm of Nister.
4. We determined affine correspondences by performing LSM. Approximate values for the affinity result from scale and direction differences

$$A = s_2/s_1 R(\phi_2 - \phi_1)$$

5. We used pairs of ACs to verify the 2-AC-algorithm above.

We obtained the following preliminary results:

- The usefulness of the 5-point algorithm was confirmed. Using a threshold of 5 pixels for the consistency check we usually found enough consensus. The difference of the relative rotation to its ground truth was in the order of a few degrees, sometimes below 1°. The angle between the estimated and the given baseline was in the order of a few degrees, never below 1°. This partly can be explained by the short baselines, being only a fraction (1/2 to 1/10) of the distance to the scene.
- The theoretical precision of the relative pose of both algorithms was comparable. We used the maximum standard deviation of the rotation angles and the base-direction. However, there was a tendency, that the solution with 2 ACs yielded more precise results. Except for weak configurations the standard deviations were below 0.5°. The precision of the relative pose from pairs of ACs varied heavily.
- The relative rotation could be recovered from 2 ACs with angular deviations in the order of a few degrees. Also here large variations appeared. Not every (correct) pair of ACs was able to recover the relative pose. This partly can be explained by the small scales and the local 3D structure, which was present. Only for very large windows and very planar regions the results came close to the true values. This indicates, that the keypoint detector needs to be able to catch large planar regions. This appears not to be the case for Lowe keypoints.

29.7.4 Bias of affine parameters

If the mapping between the images is known and smooth we are able to compare the estimated local affinity with the one derived from the known mapping. We can expect the deviations increase with increasing window size and increasing deviation from an ideal affinity. We now investigate the dependency empirically.

We choose a real image where there are no depth discontinuities in the scene, here the graffiti image. We use it to generate two noisy images with known perspective distortion.

For 21 manually selected points we determined the corresponding coordinates. At these positions we determined the local affinity $\hat{\mathsf{A}}(\hat{\boldsymbol{a}})$ and compared it with the local affinity $\mathsf{A}(\boldsymbol{a})$ derived from the Jacobian of the perspective mapping. As measure for the similarity of the two affinities we use the Mahalanobis distance

$$\underline{m}^2 = (\hat{\underline{\boldsymbol{a}}} - \boldsymbol{a})^\mathsf{T} \Sigma_{aa}^{-1} (\hat{\underline{\boldsymbol{a}}} - \boldsymbol{a}) \sim \chi_6^2 \tag{29.225}$$

which is χ_6^2-distributed if the mapping is an ideal affinity. The covariance matrix Σ_{aa} is taken from the LSM. Hence, this test statistic should be smaller than the critical value $k = k(S)$, say for a significance level of 99.9%, $k = 22.46$. For easier interpretation we give $m' = \sqrt{m^2/6}$ and compare it with $k' = \sqrt{k/6} = 4.7$, since then the distance can be interpreted similar to normally distributed test statistics.

The table contains the median and maximum values m' for different window sizes and perspective distortions. Given the radius (or scale) the window-size is $10r \times 10r$. The perspective distortion is realized by mapping the rectangle with the corner coordinates $\boldsymbol{c}_i, i = 1, 2, 3, 4$ to a quadrangle with the four corners

$$\boldsymbol{c}_i \odot (1 + d\,\boldsymbol{\varepsilon}) \quad \text{with} \quad \boldsymbol{\varepsilon} \in \mathcal{U}^2(0, 1)\,. \tag{29.226}$$

The coordinates refer to the centre of the window. Hence, the coordinates are randomly enlarged by a factor $(1 + d)$ approximately.

Table 29.1: Maximum test statistic $m' = \sqrt{\chi^2/6}$ as a function of the window size $10r \times 10r$ and the degree d of perspective distortion. Critical value $k' = 4.7$. The values result from 10 samples each

$r\backslash d$	0.0		0.1		0.2		0.4	
	median	max	median	max	median	max	median	max
2.0	1.49	3.36	3.90	9.98	4.64	12.79	5.30	17.18
3.0	1.48	3.01	6.77	18.15	8.99	25.31	10.80	31.83
4.0	1.25	2.75	10.42	24.50	12.85	41.31	17.40	71.63
5.0	1.21	3.52	14.01	36.55	19.87	61.99	23.93	124.16

In case of no perspective distortion, the estimated affinity appears to be consistent with the true affinity. The maximum test statistics keep below the critical value.

With increasing distortion and increasing window the estimated affinity shows significant bias when compared with the affinity at the centre point derived from the Jacobian of the perspective mapping.

In the following table we provide the absolute changes (RMSE) for the affine parameters and the shifts. The absolute biases decrease with larger windows, especially the bias of

Table 29.2: Root mean square errors s_A =RMSE$_A$ and s_b =RMSE$_b$ for the affine parameters $[a_1 ..., a_4]$ and the shifts $[a_5, a_6]$ as a function of the window size $10r \times 10r$ and the degree d of perspective distortion, derived from 50 samples each

$r\backslash d$	0.0		0.1		0.2		0.4	
	s_A [1]	s_b [pel]	s_A [1]	s_b [pel]	s_A [1]	s_b [pel]	s_A [1]	s_b [pel]
2.0	0.0644	0.1940	0.0688	0.3634	0.0838	0.3665	0.1221	0.3889
3.0	0.0209	0.1051	0.0302	0.3169	0.0533	0.3159	0.0916	0.3188
4.0	0.0089	0.0553	0.0231	0.2966	0.0434	0.3028	0.0798	0.2888
5.0	0.0058	0.0474	0.0230	0.2965	0.0458	0.2919	0.0891	0.2791

the affinities, while the bias of the positions only decrease for zero perspectivity, as to be expected, while being a bit below 1/3 of a pixel independent of the window size and the distortion.

We will repeat the experiment by comparing the estimated affinities with the average affinity in a given window.

30 Surface Slope from Affine Matches

We discuss how to derive the local slope of the surface from affine matches for rectified and non-rectified images. The methods are rigorous, in case the surface is locally planar and the perspective distortions are small, which usually is an acceptable assumption, except at depth or slow discontinuities.

30.1 Surface Slope from Affine Matches

Affine matches allow to recover the local slope at the observed scene point. We derive expressions to determine the surface normal for rectified and for non-rectified images.

30.1.1 Geometric models for affinities

We address two geometric models for deriving surface slope:

1. The first model assumes rectified calibrated images: the two cameras have projection matrices

$$\mathsf{P}' = [I_3 \mid \mathbf{0}], \qquad \mathsf{P}'' = [I_3 \mid -\boldsymbol{B}] \quad \text{with} \quad \boldsymbol{B} = \begin{bmatrix} B_X \\ 0 \\ 0 \end{bmatrix} \tag{30.1}$$

   Assuming rectified images an affine transformation in the direction of the first coordinate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x'' + a_2 y'' \\ y'' \end{bmatrix} \tag{30.2}$$

   This model is rigorous in case the scene is locally planar. In case the surface plane is given by

$$Z = A_0 + A_1 X + A_2 Y, \qquad \mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ -1 \\ A_0 \end{bmatrix} \tag{30.3}$$

   the three parameters a_i correspond to the slopes A_1 and A_2 of the plane in X- and Y-direction and the depth A_0.

2. The second model assumes non-rectified images: the two cameras are assumed to have projection matrices

$$\mathsf{P}' = [I_3 \mid \mathbf{0}], \qquad \mathsf{P}'' = R[I_3 \mid -\boldsymbol{B}] \quad \text{with} \quad \boldsymbol{B} = \begin{bmatrix} B_X \\ B_Y \\ B_Z \end{bmatrix} \tag{30.4}$$

Assuming an *affinity* with $U = 6$ unknown parameters

$$\boldsymbol{y} = \mathsf{A}\boldsymbol{z} + \boldsymbol{a} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \tag{30.5}$$

as the first order approximation of all more general smooth (differentiable) geometric transformations. However, the parameters a_i cannot be chosen arbitrary, but have to fulfil three constraints, since the local affinity has only three degrees of freedom.

30.1.2 Slope from rectified images

We start from the homography between the two images

$$\mathbf{x}'' = \mathsf{P}''\mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\overline{\mathsf{Q}'}^{\mathsf{T}}\mathbf{x}' = \begin{bmatrix} A_0 + B_X A_1 & B_X A_2 & -cB_X \\ 0 & A_0 & 0 \\ 0 & 0 & A_0 \end{bmatrix} \mathbf{x}' \tag{30.6}$$

where the image coordinates refer to the principle point. Therefore we have the three parameters of the 1D affinity

$$a_0 = -\frac{cB_X}{A_0}\,, \qquad a_1 = \frac{A_0 + B_X A_1}{A_0}\,, \qquad a_2 = \frac{B_X A_2}{A_0} \tag{30.7}$$

and inversely

$$A_0 = -\frac{c}{a_0}B_X\,, \qquad A_1 = -\frac{c}{a_0}\,(a_1 - 1)\,, \qquad A_2 = -\frac{c}{a_0}a_2\,. \tag{30.8}$$

The Jacobian is given by

$$\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{a}} = \frac{c}{a_0^2}\begin{bmatrix} B_X & 0 & 0 \\ a_1 - 1 & -a_0 & 0 \\ a_2 & 0 & -a_0 \end{bmatrix}. \tag{30.9}$$

30.1.3 Slope from non-rectified images

We can derive the normal of the plane following Eichhardt and Chetverikov (2018, eq. 2). We use $\boldsymbol{x} = \boldsymbol{c}(\mathbf{x}) = \boldsymbol{x}_0/x_h$ and thus $\mathrm{d}\boldsymbol{x} = J_c(\mathbf{x})\mathrm{d}\mathbf{x}$ (see Förstner and Wrobel (2016, eq. (10.33)), with

$$J_c(\mathbf{x}) = \frac{1}{x_h^2}[x_h I_2 \mid -\boldsymbol{x}_0] \tag{30.10}$$

We obtain from

$$\boldsymbol{x}' = \boldsymbol{c}(\boldsymbol{Z}) \quad \text{and} \quad \boldsymbol{x}'' = \boldsymbol{c}(R(\boldsymbol{Z} - \boldsymbol{B})) \tag{30.11}$$

the differentials

$$\mathrm{d}\boldsymbol{x}' = J_c(\mathbf{x}')\mathrm{d}\boldsymbol{Z}({}^m\boldsymbol{X}) \quad \text{and} \quad \mathrm{d}\boldsymbol{x}'' = J_c(\mathbf{x}'')R\mathrm{d}\boldsymbol{Z}({}^m\boldsymbol{X}) \tag{30.12}$$

Here we assume the Z_3 coordinates depend on the map-coordinates ${}^m\boldsymbol{X}$ via

$$Z = A_0 + A_1{}^mX_1 + A_2{}^mX_2 \quad \text{or} \quad \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ A_0 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ A_1 & A_2 \end{bmatrix}}_{J_Z} \begin{bmatrix} {}^mX_1 \\ {}^mX_2 \end{bmatrix}. \tag{30.13}$$

Hence, the differentials are

$$\mathrm{d}\boldsymbol{x}' = J_1\mathrm{d}{}^mX \quad \text{and} \quad \mathrm{d}\boldsymbol{x}'' = J_2\mathrm{d}{}^mX \tag{30.14}$$

with

$$J_1 = J_c(\mathbf{x}')\; J_Z \quad \text{and} \quad J_2 = J_c(\mathbf{x}'')\; R\; J_Z \tag{30.15}$$

The local affinity between the image coordinates is $\boldsymbol{y} = \boldsymbol{x}'' + A(\boldsymbol{y} - \boldsymbol{x}')$, hence

$$\mathrm{d}\boldsymbol{x}' = A\mathrm{d}\boldsymbol{x}'' \quad \text{with} \quad A = J_2 J_1^{-1} . \tag{30.16}$$

We explicitly have

$$J_c(\mathbf{x}')\ J_Z\ A = J_c(\mathbf{x}'')\ R\ J_Z \tag{30.17}$$

With $\boldsymbol{n} = [A_1, A_2]^\mathsf{T}$ and thus $J_Z = [I \mid \boldsymbol{n}]^\mathsf{T}$ we therefore find a relation linear in $\boldsymbol{n}$

$$J_c(\mathbf{x}')\ \begin{bmatrix} A \\ \boldsymbol{n}^\mathsf{T} A \end{bmatrix} = J_c(\mathbf{x}'')\ R\ \begin{bmatrix} I_2 \\ \boldsymbol{n}^\mathsf{T} \end{bmatrix} . \tag{30.18}$$

This reads as

$$\frac{1}{x_h'^2}(x_h' A - \boldsymbol{x}_0' \boldsymbol{n}^\mathsf{T} A) = (J_c(\mathbf{x}'')\ R)_{:,1:2} + (J_c(\mathbf{x}'')\ R)_{:,3}\boldsymbol{n}^\mathsf{T} \tag{30.19}$$

or

$$\frac{1}{x_h'^2} x_h' \mathrm{vec}(A) - \frac{1}{x_h'^2}(A^\mathsf{T} \otimes \boldsymbol{x}_0')\boldsymbol{n} = \mathrm{vec}\left((J_c(\mathbf{x}'')\ R)_{:,1:2}\right) + (I_2 \otimes (J_c(\mathbf{x}'')\ R)_{:,3})\boldsymbol{n} \tag{30.20}$$

Thus

$$\left(\frac{1}{x_h'^2}(A^\mathsf{T} \otimes \boldsymbol{x}_0') + (I_2 \otimes (J_c(\mathbf{x}'')\ R)_{:,3})\right)\boldsymbol{n} = \frac{1}{x_h'}\mathrm{vec}(A) - \mathrm{vec}\left((J_c(\mathbf{x}'')\ R)_{:,1:2}\right) . \tag{30.21}$$

This is an overdetermined 4×2 equation system for the 2-vector $\boldsymbol{n}$. Since A may not be arbitrary, but has to fulfil two constraints, we can expect the equation system to be consistent.

The two constraints restrict the rotation and the scaling of the affinity A.

31 Local Affinity of a Homography and its Approximation

Image matching may be based on affine matches. The estimated affinity depends on local windows with a size depending on the scale e.g., of a keypoint detector. In case of locally planar scenes they approximate the Jacobian of a homography. On the other hand, the scale and rotation parameters provided by a keypoint detector, do not encode the local shears, but may be used as approximation for the local affinity, possibly refined by some intensity-based matching. We discuss the Jacobian of a homography, provide methods to determine the scaled rotation part of the affinity, and a criterion to identify the lack of shears. This enables to empirically determine the accuracy of the scale and rotation parameters derived from Lowe keypoints.

31.1 Task

Given is a homography $\chi(\mathbf{x}) \mapsto y(\mathbf{y})$ such that $\mathbf{y} = \mathsf{H}\mathbf{x}$. Derive the Jacobian for the non-homogeneous coordinates, i.e., the local affinity at a given point $\chi(\boldsymbol{x})$

$$A = \left.\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right|_x , \tag{31.1}$$

provide methods to determine the scaled rotation part of the affinity, and a criterion to identify the lack of shears. This enables to empirically determine the accuracy of the scale and rotation parameters derived from Lowe keypoints.

31.2 Derivation

If the homography is described using the row vectors or using its partitioning

$$\mathsf{H} = \begin{bmatrix} \mathbf{h}_1^\mathsf{T} \\ \mathbf{h}_2^\mathsf{T} \\ \mathbf{h}_3^\mathsf{T} \end{bmatrix} = \begin{bmatrix} A_H & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & s \end{bmatrix} , \tag{31.2}$$

the non-homogenous coordinates of y read as

$$\boldsymbol{y}(\boldsymbol{x}) = \frac{\begin{bmatrix} \mathbf{h}_1^\mathsf{T}\mathbf{x} \\ \mathbf{h}_2^\mathsf{T}\mathbf{x} \end{bmatrix}}{\mathbf{h}_3^\mathsf{T}\mathbf{x}} \quad \text{with} \quad \mathbf{x} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} . \tag{31.3}$$

With the symbolic package of Matlab the Jacobian yields

$$A = \left.\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right|_x \tag{31.4}$$
$$\left(\begin{array}{cc} \frac{H_{1,1}}{H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2} - \frac{H_{3,1}\,(H_{1,3}+H_{1,1}\,x_1+H_{1,2}\,x_2)}{(H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2)^2} & \frac{H_{1,2}}{H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2} - \frac{H_{3,2}\,(H_{1,3}+H_{1,1}\,x_1+H_{1,2}\,x_2)}{(H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2)^2} \\ \frac{H_{2,1}}{H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2} - \frac{H_{3,1}\,(H_{2,3}+H_{2,1}\,x_1+H_{2,2}\,x_2)}{(H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2)^2} & \frac{H_{2,2}}{H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2} - \frac{H_{3,2}\,(H_{2,3}+H_{2,1}\,x_1+H_{2,2}\,x_2)}{(H_{3,3}+H_{3,1}\,x_1+H_{3,2}\,x_2)^2} \end{array} \right)$$

With the homogeneous coordinates of the transformed point

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{y}_0 \\ y_h \end{bmatrix} = \begin{bmatrix} H_{1,3} + H_{1,1}\,x_1 + H_{1,2}\,x_2 \\ H_{2,3} + H_{2,1}\,x_1 + H_{2,2}\,x_2 \\ H_{3,3} + H_{3,1}\,x_1 + H_{3,2}\,x_2 \end{bmatrix} \tag{31.5}$$

we find the short version

$$\boxed{A = \frac{1}{y_h^2}(y_h A_H - \boldsymbol{y}_0 \boldsymbol{p}^\mathsf{T})\,.} \tag{31.6}$$

Observe it also can be derived by cantering the coordinate system into $\boldsymbol{y}$ using

$${}^c\mathbf{y} = \mathsf{T}\mathbf{y} \quad \text{with} \quad \mathsf{T} = \begin{bmatrix} I_2 & -\boldsymbol{y} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{31.7}$$

yielding the homography

$${}^c\mathbf{y} = {}^c\mathsf{H}\mathbf{x} \quad \text{with} \quad {}^c\mathsf{H} = \mathsf{T}\mathsf{H} = \begin{bmatrix} I_2 & -\boldsymbol{y} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} A_H & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & s \end{bmatrix} = \begin{bmatrix} A_H - \boldsymbol{y}\boldsymbol{p}^\mathsf{T} & \boldsymbol{t} - s\boldsymbol{y} \\ \boldsymbol{p}^\mathsf{T} & s \end{bmatrix} \tag{31.8}$$

From (31.6) we can conclude that, if $\boldsymbol{y}_0 = \mathbf{0}$, we just need to extract the affinity of ${}^c\mathsf{H}$ and divid by y_h. Since ${}^c\boldsymbol{y} = \mathbf{0}$ we arrive at

$$A = \frac{1}{y_h}(A_{\,{}^cH}) = \frac{1}{y_h}(A - \boldsymbol{y}\boldsymbol{p}^\mathsf{T})\,. \tag{31.9}$$

31.3 Affinity for observed slanted plane

Let a plane be given by its homogeneous coordinates $\mathbf{A} = [\boldsymbol{A}_h^\mathsf{T}, A_0]^\mathsf{T}$ and observed by an image pair in normal pose, i.e., $R = I_3$ and $\boldsymbol{B} = \boldsymbol{e}_1$. Then the homography

$${}^c\mathbf{x}' = \mathsf{H}{}^c\mathbf{x}'' \quad \text{with} \quad \mathsf{H} = I + \frac{\boldsymbol{T}\boldsymbol{A}_h^\mathsf{T}}{A_0} = \begin{bmatrix} A_H & \boldsymbol{t} \\ \boldsymbol{p}^\mathsf{T} & s \end{bmatrix} \tag{31.10}$$

can be used to derive the local affinity

$$A = \frac{1}{y_h^2}(y_h A_H - \boldsymbol{y}_0 \boldsymbol{p}^\mathsf{T}) \tag{31.11}$$

with $\boldsymbol{y}_0$ from

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{y}_0 \\ y_h \end{bmatrix} = \mathsf{H}\mathbf{x}\,, \tag{31.12}$$

where $\mathbf{x}$ is the point of linearization.

In our special case we assume the plane is

$$Z = H + XZ_X + YZ_Y \quad \text{or} \quad \mathbf{A} = \begin{bmatrix} Z_X \\ Z_Y \\ -1 \\ H \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ -1 \end{bmatrix} \\ H \end{bmatrix}. \tag{31.13}$$

The homography is

$$\mathsf{H} = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} + \frac{1}{H} \begin{bmatrix} \boldsymbol{e}_1^{[2]} \\ 0 \end{bmatrix} [\boldsymbol{n}^\mathsf{T} \mid -1] = \begin{bmatrix} I_2 + \boldsymbol{e}_1^{[2]} \boldsymbol{n}^\mathsf{T}/H & -\boldsymbol{e}_1^{[2]}/H \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{31.14}$$

This is an affinity. Hence the Jacobian is independent on the position

$$A = \frac{\partial \boldsymbol{x}''}{\partial \boldsymbol{x}'} = I_2 + \boldsymbol{e}_1^{[2]} \boldsymbol{n}^\mathsf{T}/H = \begin{bmatrix} 1 + Z_X/H & Z_Y/H \\ 0 & 1 \end{bmatrix}. \tag{31.15}$$

31.4 Code for Jacobian

The Matlab code for determining the Jacobian is

```
%% local affinity from homography at point
x=sym('x',[2,1],'real')
y=sym('y',[2,1],'real')
H=sym('H',[3,3],'real')
y= H(1:2,:)*[x;1]./([1;1]*H(3,:)*[x;1]);
jacobian(y,x)
```

31.5 Partitioning of an affinity

We assume $\tilde{A}_i \in \mathbb{R}^{2\times 2}$ matrix locally approximate the homography $\tilde{H}_i \in \mathbb{R}^{3\times 3}$. The goal of comparing SIFT directions could be to determine the rotation component $\tilde{R}$ of the affinity $\tilde{A}_i$ and compare it to the angle between the directions of corresponding keypoints.

We address three alternatives for determining the rotational component of $\tilde{A}$:

1. a QR-decomposition,
2. a SVD-decomposition, and
3. an exponential decomposition.

Rotation from QR-decomposition of an affinity A. Assuming the affinity is a concatenation of a shear matrix S and a subsequent rotation with R

$$A = RS \tag{31.16}$$

the classical QR-decomposition is defined as

$$R_{\mathrm{qr,A}} := R \quad \text{with} \quad [R, S] = \mathrm{qr}(A)\,. \tag{31.17}$$

In case the affinity is defined by the reverse sequence, i.e.,

$$A = SR \tag{31.18}$$

the QR decomposition of the transposed needs to be taken

$$R_{\mathrm{qr,A}^\mathsf{T}} := R^\mathsf{T} \quad \text{with} \quad [R, S] = \mathrm{qr}(A^\mathsf{T})\,. \tag{31.19}$$

If there are no shears, i.e., the shear matrix is a scaled unit matrix, the two rotations $R_{\mathrm{qr,A}}$ and $R_{\mathrm{qr,A}^\mathsf{T}}$ are the same, otherwise they differ.

Rotation from SVD-decomposition of A. An alternative way to derive the rotation component uses the matrix exponential. Let us assume, the affinity is decomposable as two rotations sandwiching a individual scaling

$$A = UDV^{\mathsf{T}} \quad \text{with} \quad D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}, \tag{31.20}$$

where the shears are represented by the rotation V and the ratio d_1/d_2. Then the SVD yields the rotation

$$R_{\mathrm{svd,A}} := UV^{\mathsf{T}} \quad \text{with} \quad [U, \mathbf{\Lambda}, V] = \mathrm{svd}(A)\,. \tag{31.21}$$

Transposing A does not change the rotation. The resulting rotation only is identical to those of the QR-decomposition if the affinity is a scaled rotation.

Rotation from an exponential decomposition The affinity A can be written as an exponential of a matrix B

$$A = \mathrm{e}^{B} \tag{31.22}$$

If the matrix B is zero, i.e. $B = 0$, the affinity is a unit transformation. We now can decompose the exponent additively in the following form

$$B = \sum_i p_i B_i \tag{31.23}$$

with the four basic 2×2 matrices

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{31.24}$$

$$B_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{31.25}$$

Hence

$$A = \mathrm{e}^{p_1 B_1 + p_2 B_2 + p_3 B_3 + p_4 B_4}\,. \tag{31.26}$$

If we take each of the summands individually, the four parameters refer to (1) scaling with $\log p_1$, (2) rotation by p_2 [rad], (4) 1st shear, namely opposite rotation of axes, and (4) 2nd shear, namely opposite scaling of axes, see Fig 31.1.

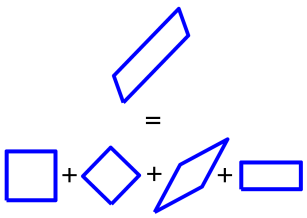


Figure 31.1: The affinity $A = \exp(B)$ may be interpreted as the combined effect of the four basic transformations scaling, rotation, the first and the second shear, here with parameters $\boldsymbol{p} = (0.2, 0.8, 0.6, 0.4)$ showing the individual transformations $\exp(p_i B_i)$. The plus-signs in the figure refer to the exponents

The rotation is given by the well-known relation

$$R = \exp(p_2 B_2)\,. \tag{31.27}$$

Furthermore, for the first shear we explicitely have

$$\exp\left(\begin{bmatrix} 0 & p_4 \\ p_4 & 0 \end{bmatrix}\right) \tag{31.28}$$

$$= \begin{bmatrix} \mathrm{e}^{-p_4/2} + \mathrm{e}^{p_4/2} & \mathrm{e}^{p_4/2} - \mathrm{e}^{-p_4/2} \\ \mathrm{e}^{-p_4/2} - \mathrm{e}^{-p_4/2} & \mathrm{e}^{-p_4/2} + \mathrm{e}^{p_4/2} \end{bmatrix} \tag{31.29}$$

$$\overset{q_4=\mathrm{e}^{p_4/2}}{=} \begin{bmatrix} q_4 + 1/q_4 & q_4 - 1/q_4 \\ q_4 - 1/q_4 & q_4 + 1/q_4 \end{bmatrix}. \tag{31.30}$$

This representation is highly symmetric. The additive terms are invariant w.r.t. the sequence of the terms. Moreover, the scaled rotation is independent on the existence of shears.

However, since the exponent of two matrices only is the product of the two matrices if they commute, i.e.

$$\exp(A + B) = \exp(A)\exp(B) \quad \text{only if} \quad AB = BA\,, \tag{31.31}$$

the interpretation of the elements in the exponent is not independent of the existence of the other elements. Only a common scaling can be exchanged with the other components, as is known from scaled rotation.

Now, we can define the rotational component using (31.27) deriving p_2 from

$$p_2 = (B(2,1) - B(1,2))/2 \quad \text{with} \quad B = \log(A) \tag{31.32}$$

where $\log(A)$ is the matrix logarithm of A.

Therefore we are able to identify the existence of shears, namely we have no shears if

$$d_s^2 = |[p_3, p_4]| = p_3^2 + p_4^2 = 0 \tag{31.33}$$

Since a scale rotation has condition number $\mathrm{cond}(sR) = 1$, also the condition number can be used to identify the lack of shears, namely if $\mathrm{cond}(A) = 1$. For not too large shears the the condition number and the degree of shears d_s^2 are approximately the same:

$$d_s^2 \approx \mathrm{cond}(A)\,. \tag{31.34}$$

31.6 Affinity and Slope of Plane

We give a relation between the condition number and the slope of a plane observed by an image pair in normal position.

The image of a sloped plane leads to scale differences s and shears a due to the tilts Z_x and Z_y of the plane along and across the base line. They have the form

$$A_s = \begin{bmatrix} 1+s & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad A_a = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \tag{31.35}$$

The combined effect is the affinity

$$A_{sa} = A_a A_s = \begin{bmatrix} 1+s & a \\ 0 & 1 \end{bmatrix} \tag{31.36}$$

Condition Number for Affinity except Scaled Rotation. The condition number of this affinity, is given by

$$c = 1 + \frac{(\sqrt{4(1+s) + t^2} + t)}{2(1+s)} t \quad \text{with} \quad t^2 = a^2 + s^2 \tag{31.37}$$

For small s and t it can be approximated by

$$c \approx 1 + t = 1 + \sqrt{a^2 + s^2} \tag{31.38}$$

neglecting higher order terms. A condition number $c = 1.5$ corresponds to $t = \sqrt{2}/3$.

Affine Parameters and Slope of Scene Plane. Assume the stereo image pair in normal position with rotation $R = I$, basis $b = [1, 0, 0]^\mathsf{T}$, and focal length $f = 1$ with coordinate system in the first camera observing a sloped plane at $[0, 0, Z_0]^\mathsf{T}$

$$Z = Z_0 + XZ_X + YZ_Y \quad \text{with} \quad Z_X = \frac{\partial Z}{\partial X}\,, \;\; Z_Y = \frac{\partial Z}{\partial Y} \tag{31.39}$$

or with homogeneous plane coordinates

$$A = [Z_X, Z_Y, -1, Z_0]^\mathsf{T} = [n^\mathsf{T}, Z_0]^\mathsf{T}\,. \tag{31.40}$$

The homography from x' to x'' is given by $x'' = Hx'$ by

$$H = I + \frac{bn^\mathsf{T}}{Z_0} = \begin{bmatrix} \frac{\mathrm{Z_X}}{Z_0} + 1 & \frac{\mathrm{Z_Y}}{Z_0} & -\frac{1}{Z_0} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{31.41}$$

This is an affinity with a Jacobian independent of the position in the image, namely

$$A = \frac{\partial x''}{\partial x'} = \begin{pmatrix} 1 + \frac{Z_X}{Z_0} & \frac{Z_Y}{Z_0} \\ 0 & 1 \end{pmatrix}. \tag{31.42}$$

Hence, we have the scale difference and the shear

$$s = \frac{\mathrm{Z_X}}{Z_0} \quad \text{and} \quad a = \frac{\mathrm{Z_Y}}{Z_0}\,. \tag{31.43}$$

If the scale difference and the shear are $s = a = 1/3$, the slope of the plane is $\operatorname{atan2}(\sqrt{2}, 3) = 25.24°$ and the condition number is $c = 1.5$, which we use as threshold for approximate scale rotations.

32 Noise Suppression with Box, Binomial, and Gaussian Filters

We derive expressions for the scale dependent decrease of the variance of white noise images when being smoothed with a box, with a Binomial, with a Gaussian filter, and partial derivatives of a Gaussian filter of up to second order.

32.1 Task

Low pass filters aim at suppressing the noise in an image. This note provides explicit expressions for the decrease in noise variance when applying box, binomial, and Gaussian filters.

32.2 Noise variances for discrete filters

The result is given in the following theorem.

Theorem 32.2.9: ***Noise variance for discrete filters.***

If the noise $\underline{n}$ in an image $\underline{g}$ has zero mean, standard deviation σ_n, and is uncorrelated, the noise $\overline{n}$ in the image

$$\overline{\underline{g}} = \underline{g} * H \quad \text{or} \quad \overline{\underline{g}}(x) = \sum_k H(k) g(x-k) \quad \text{or} \quad \overline{\underline{g}}(x) = \int_t H(t) g(x-t) \tag{32.1}$$

has standard deviation

$$\sigma_{\overline{n}} = c(H) \cdot \sigma_n \tag{32.2}$$

with

$$c^2(H) = \sum_k H^2(k) \quad \text{or} \quad c^2 = \int_t H^2(t)\,. \tag{32.3}$$

For 1D- and 2D-images and box filter $H = R_m$ and binomial filter $H = B_m$ the ratios $c_m = \sigma_{\overline{n}}/\sigma_n$ are given in the table as function of the width of the filters.

Table 32.1: Rigorous expressions for the noise standard deviation for box and binomial filters

$c_m = \frac{\sigma_{\overline{n}}}{\sigma_n}$	1D	2D
Box R_m	$\frac{1}{\sqrt{m}}$	$\frac{1}{m}$
Binomial B_m	$\sqrt{\frac{1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \cdots \cdot 2m}}$	$\frac{1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \cdots \cdot 2m}$

For large m these expressions can be approximated by the values in the following table.

Table 32.2: Approximate expressions (large m) for the noise standard deviation for box and binomial filters

$c_m \approx \frac{\sigma_{\overline{n}}}{\sigma_n}$	1D	2D
Box R_m	$\frac{1}{\sqrt{m}} = \frac{1}{\sqrt[4]{12}\sqrt{\sigma_{R_m}}} = 0.5373\frac{1}{\sqrt{\sigma_{R_m}}}$	$\frac{1}{m} = \frac{1}{\sqrt{12}\,\sigma_{R_m}} = 0.2887\frac{1}{\sigma_{R_m}}$
Binomial B_m	$\frac{1}{\sqrt[4]{\pi m}} = \frac{1}{\sqrt[4]{4\pi}\sqrt{\sigma_{B_m}}} = 0.5311\frac{1}{\sqrt{\sigma_{B_m}}}$	$\frac{1}{\sqrt{\pi m}} = \frac{1}{\sqrt{4\pi}\,\sigma_{B_m}} = 0.2821\frac{1}{\sigma_{B_m}}$

using the widths (spatial standard deviation)

$$\sigma_{R_m} = \sqrt{\frac{m^2-1}{12}} \tag{32.4}$$

$$\sigma_{B_m} = \sqrt{\frac{m}{4}} \tag{32.5}$$

of the filters.

32.3 Derivation

The equations for the box filter are obtained easily.

For the binomial filter we show that

$$c_m^2 = \frac{1}{(2^m)^2}\sum_{k=0}^{m}\left[\binom{m}{k}\right]^2 = \frac{1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \cdots \cdot 2m} = \frac{(2m-1)!!}{(2m)!!} \tag{32.6}$$

with the double factorial for odd n and even m

$$n!! = n(n-2)\dots 1 \quad \text{and} \quad m!! = m(m-2)\dots 2\,. \tag{32.7}$$

This expression is just the sum of the squares of the elements of the kernel B_m, needed for error propagation.

Remark: Proofwiki contains proofs for the relation:

$$\sum_{k=0}^{m} \binom{m}{k}^2 = \binom{2m}{m} \tag{32.8}$$

see https://proofwiki.org/wiki/Sum_of_Squares_of_Binomial_Coefficients. Hence, we obtain

$$c_m^2 = \frac{1}{(2^m)^2} \sum_{k=0}^{m} \binom{m}{k}^2 = \frac{1}{4^m} \frac{2m!}{m!m!} = \frac{2m!}{(2^m m!)(2^m m!)} = \frac{\prod_k (2k)\ \prod_k 2k-1}{\prod_k 2k \prod_k 2k} = \frac{\prod_k 2k-1}{\prod_k 2k}\,. \tag{32.9}$$

q.e.d. ◇

We explicitly obtain for $m = 2, 3, 4, 5, 6$

$$m=1: \quad \left(\frac{1}{2}\right)^2 (1^2+1^2) = \frac{1}{2} = 0.5 \tag{32.10}$$

$$m=2: \quad \left(\frac{1}{4}\right)^2 (1^2+2^2+1^2) = \frac{3}{8} = \frac{1\cdot 3}{2\cdot 4} \tag{32.11}$$

$$m=3: \quad \left(\frac{1}{8}\right)^2 (1^2+3^2+3^2+1^2) = \frac{5}{16} = \frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6} \tag{32.12}$$

$$m=4: \quad \left(\frac{1}{10}\right)^2 (1^2+4^2+6^2+4^2+1^2) = \frac{35}{128} = \frac{1\cdot 3\cdot 5\cdot 7}{2\cdot 4\cdot 6\cdot 8} \tag{32.13}$$

$$m=5: \quad \left(\frac{1}{32}\right)^2 (1^2+5^2+10^2+10^2+5^2+1^2) = \frac{63}{256} = \frac{1\cdot 3\cdot 5\cdot 7\cdot 9}{2\cdot 4\cdot 6\cdot 8\cdot 10}$$

$$m=6: \quad \left(\frac{1}{64}\right)^2 (1^2+6^2+15^2+20^2+15^2+6^2+1^2) = \frac{231}{1024} = \frac{1\cdot 3\cdot 5\cdot 7\cdot 9\cdot 11}{2\cdot 4\cdot 6\cdot 8\cdot 10\cdot 12}$$

The coeffient c for the 2D-binomial can be written as

$$c_m^2 = \frac{(2m)!}{(2^m m!)^2} \tag{32.14}$$

With Stirlings formula for $m!$

$$m! \approx \left(\frac{m}{\mathrm{e}}\right)^m \sqrt{2\pi m} \tag{32.15}$$

we obtain the approximation

$$c_m^2 = \frac{(\frac{2m}{\mathrm{e}})^{2m} \sqrt{2\pi \cdot 2m}}{2^{2m} \cdot (\frac{m}{\mathrm{e}})^{2m} \cdot 2\pi m} = \frac{1}{\sqrt{\pi m}} \tag{32.16}$$

This is valid for large m.

It can also be obtained by the continuous approximation of the binomial B_m namely the Gaussian function G_σ

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right) \tag{32.17}$$

We have

$$
\begin{aligned}
c^2(G(x;\sigma)) &= \int_{x=-\infty}^{\infty} G^2(x;\sigma)\,\mathrm{d}x && (32.18)\\
&= \int_{x=-\infty}^{\infty} \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2}{\sigma^2}\right)\,\mathrm{d}x && (32.19)\\
&= \int_{x=-\infty}^{\infty} \frac{1}{2\sqrt{\pi}\sigma}\frac{1}{\sqrt{2\pi(\sigma/\sqrt{2})^2}}\exp\left(-\frac{1}{2}\frac{x^2}{(\sigma/\sqrt{2})^2}\right)\,\mathrm{d}x && (32.20)\\
&= \frac{1}{2\sqrt{\pi}\sigma}\underbrace{\int_{x=-\infty}^{\infty} \frac{1}{\sqrt{2\pi(\sigma/\sqrt{2})^2}}\exp\left(-\frac{1}{2}\frac{x^2}{(\sigma/\sqrt{2})^2}\right)\,\mathrm{d}x}_{1} && (32.21)\\
&= \frac{1}{2\sqrt{\pi}\sigma}\,. && (32.22)
\end{aligned}
$$

With (32.5) we obtain 32.16.

32.4 Variances of Gaussian-filtered noise images

This section collects the effect of Gaussian related filters, including partial derivatives, onto the degree of smoothing white noise.

Let the central 1D- and 2D-Gaussian with width σ be

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right) \tag{32.23}$$

and

$$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}\right) = G(x;\sigma)\,G(y;\sigma)\,. \tag{32.24}$$

The coefficients $c^2 = \sigma_{\tilde{n}}^2/\sigma_n^2$ for Gaussian related filters result from (32.3). We have for 1D signals the factors

$$
\begin{aligned}
c^2(G(x;\sigma)) &= \frac{1}{2\sqrt{\pi}}\frac{1}{\sigma} &&\approx 0.2820947918\frac{1}{\sigma} && (32.25)\\
c^2(G_x(x;\sigma)) &= \frac{1}{4\sqrt{\pi}}\frac{1}{\sigma^3} &&\approx 0.1410473959\frac{1}{\sigma^3} && (32.26)\\
c^2(G_{xx}(x;\sigma)) &= \frac{3}{8\sqrt{\pi}}\frac{1}{\sigma^5} &&\approx 0.2115710938\frac{1}{\sigma^5}\,. && (32.27)
\end{aligned}
$$

As an example, compare the result in (32.25), which is $c^2(G(x;\sigma)) = 1/(2\sqrt{\pi}\sigma)$ with the result in Table 32.2 for the 1D Binomial filter, which is $c_m^2 = 1/(2\sqrt{\pi}\sigma_{B_m})$ when using $m = (2\sigma_{B_m})^2$ from (32.5), see the derivation of (32.22).

For 2D signals we have the factors

$$
\begin{aligned}
c^2(G(x,y;\sigma)) &= \frac{1}{4\pi}\frac{1}{\sigma^2} &&\approx 0.07957747152\frac{1}{\sigma^2} && (32.28)\\
c^2(G_x(x,y;\sigma)) &= \frac{1}{8\pi}\frac{1}{\sigma^4} &&\approx 0.03978873576\frac{1}{\sigma^4} && (32.29)\\
c^2(G_{xx}(x,y;\sigma)) &= \frac{3}{16\pi}\frac{1}{\sigma^6} &&\approx 0.05968310364\frac{1}{\sigma^6} && (32.30)\\
c^2(G_{xy}(x,y;\sigma)) &= \frac{1}{16\pi}\frac{1}{\sigma^6} &&\approx 0.01989436788\frac{1}{\sigma^6} && (32.31)
\end{aligned}
$$

Observe, we have the relations

$$\begin{aligned} c^2(G(x,y;\sigma)) &= c^4(G(x;\sigma)) && (32.32)\\ c^2(G_x(x,y;\sigma)) &= c^2(G(x;\sigma))\, c^2(G_x(x;\sigma)) && (32.33)\\ c^2(G_{xx}(x,y;\sigma)) &= c^2(G(x;\sigma))\, c^2(G_{xx}(x;\sigma)) && (32.34)\\ c^2(G_{xy}(x,y;\sigma)) &= c^4(G_x(x;\sigma))\,. && (32.35) \end{aligned}$$

For the Mexican hat

$$\Delta G(x,y;\sigma) = G_{xx}(x,y;\sigma) + G_{yy}(x,y;\sigma) \tag{32.36}$$

we obtain the variance

$$c^2(\Delta G(x,y;\sigma)) = \frac{3}{8\pi}\frac{1}{\sigma^6}\,. \tag{32.37}$$

Part V

Technical Notes on Geometry

33 Circles and Circular Segments

We collect representations for circles and circle segments and their estimation from given points. Circle segments may be useful for geometric reasoning, especially if the angular support is small, since then the covariance matrix of the circle segment parameters have a much lower condition number than the covariance matrix of the corresponding circle.

33.1 Summary

An estimated circle segment s refers to a certain position of the circle, which can be freely chosen. As the circle, it has three degrees of freedom namely the position across the circle, the direction of the tangent and the curvature. It is represented as

$$s: \qquad \{x, y, \phi, \kappa; \mathbb{D}([q, \phi, \kappa])\}\,. \tag{33.1}$$

The tangent coordinate system is provided by (x, y, ϕ). In addition, we need the curvature κ. The uncertainty is represented by the covariance matrix $\mathbb{D}([q, \phi, \kappa])$.

It can be derived from the best fitting circle, which directly can be determined from the given points without showing a bias in the radius.

The note collects representations for the circle and circle segments and their estimation from given points.

33.2 Representation of the circle

33.2.1 The implicit form

A general circle with centre $\boldsymbol{x}_0 = [x_0, y_0]^\mathsf{T}$ and radius r is represented by the implicit function

$$(x - x_0)^2 + (y - y_0)^2 - r^2 = 0. \tag{33.2}$$

It cannot represent straight lines. The following representation with the homogeneous coordinates

$$\mathbf{Y} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = T \begin{bmatrix} x^2 + y^2 \\ x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \tag{33.3}$$

we have

$$A(x^2 + y^2) + Bx + Cy + D = 0\,. \tag{33.4}$$

In case we have a circle we have $A \neq 0$ and can write (33.2)

$$(x^2 + y^2) - 2x_0 x - 2y_0 y + (x_0^2 + y_0^2 - r^2) = 0 \tag{33.5}$$

which yields

$$x_0 = -\frac{B}{2A}, \qquad y_0 = -\frac{C}{2A}, \qquad r = \sqrt{\frac{B^2 + C^2 - 4AD}{4A^2}}. \tag{33.6}$$

In case we have a straight line we have $A = 0$ and its homogeneous coordinates are

$$\mathbf{l} = \begin{bmatrix} B \\ C \\ D \end{bmatrix}. \tag{33.7}$$

33.2.2 On the sign of the radius

The basic equation of the circle is invariant to the sign of the radius. However the curvature of the circle depends on the chosen algebraic form.

The curvature of a curve is positive if its tangent rotates counter clockwise, otherwise it is negative, though there are opposite definitions.

The unit circle is given parametrically as

$$\boldsymbol{x}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} r\cos t \\ r\sin t \end{bmatrix} \tag{33.8}$$

In case $r > 0$ the point $\boldsymbol{x}$ runs counter clockwise. E.g. at $t = 0$ or $\boldsymbol{x} = [1; 0]$ it runs upwards. If $r < 0$ the point $\boldsymbol{x}(t)$ runs clockwise.

The direction of the tangent vector is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -r\sin t \\ r\cos t \end{bmatrix} \tag{33.9}$$

E.g. at $t = 0$ or $\boldsymbol{x} = [1; 0]$ the tangent direction is $[0; 1]$, confirming the direction of the motion of the point.

The curvature is given by (see https://mathworld.wolfram.com/Curvature.html)

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \tag{33.10}$$

In our case it is

$$\kappa = \frac{(-r\sin t)(-r\sin t) - (r\cos t)(-r\cos t)}{(r^2\sin^2 t + r^2\cos^2 t)^{3/2}} = \frac{1}{r} \tag{33.11}$$

Thus, the curvature and the radius have the same sign.

33.3 Estimating the circle parameters from points

Given the N points $(x_n, y_n), n = 1, ..., N$ having weights w_n several direct solutions for estimating the circle parameters $\boldsymbol{\theta} = [x_0, y_0, r]^\mathsf{T}$ are available, see Förstner and Wrobel (2016, Sect. 4.9.2.5). The covariance matrix of the circle parameters is

$$D\left(\begin{bmatrix} \hat{x}_0 \\ \hat{y}_0 \\ \hat{r} \end{bmatrix}\right) = \Sigma_{\hat{\theta}\hat{\theta}} = \frac{\sigma_0^2}{\sum_n w_n} \begin{bmatrix} \overline{c^2} & \overline{cs} & \overline{c} \\ \overline{cs} & \overline{s^2} & \overline{s} \\ \overline{c} & \overline{s} & \overline{w} \end{bmatrix}^{-1} \tag{33.12}$$

with the abbreviations

$$\overline{c} = \frac{\sum_n w_n c_n}{\sum_n w_n}, \quad \overline{c^2} = \frac{\sum_n w_n c_n^2}{\sum_n w_n}, \quad \dots. \tag{33.13}$$

and

$$w_n = \frac{1}{\sigma_n^2}, \quad c_n = \cos(\hat{\alpha}_n) = \frac{\hat{x}_n - \hat{x}_0}{\hat{r}} \quad \text{and} \quad s_n = \sin(\hat{\alpha}_n) = \frac{\hat{y}_n - \hat{y}_0}{\hat{r}}. \tag{33.14}$$

33.3.1 Local representation if circular segment

We now choose the point on the circle as *centre* of the circle segment. We are free to choose the point as long it represents the point distribution on the segment.

This can be achieved by transforming the original points into the centroid, see Fig. 33.1. If we, at the same time scale the points, such that the coordinates on an average are close to one, we obtain a task specific conditioning of the given points, which anyway is necessary.

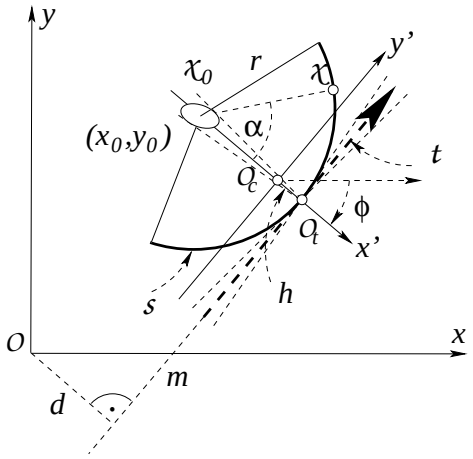


Figure 33.1: Circle segment s derived from circle $\mathcal{C}(x_0, y_0, r)$. The centroid system with centre O_c together with the major axes of the given points is used for estimating the circle and assumed to be fixed. The circle segment is represented by (x_t, y_t, ϕ, κ), where (x_t, y_t) is the origin of the tangent system, parallel to the centroid system, ϕ the direction of the ${}^t x$-axis and κ the curvature. The shift h between the two coordinate systems is uncertain. We assume the tangent system to be fixed, but the tangent to be uncertain, thus close to the ${}^t y$ axis, with two degrees of freedom: a shift q along the ${}^t x$ axis, with observed value 0 and standard deviation $\sigma_q = \sigma_h$ and a rotation with ${}^t\phi = 0$ and standard deviation σ_ϕ. In addition we have the uncertainty of the radius r or the curvature $\kappa = 1/r$.

This transformation can be determined from

$$\boldsymbol{x}'_n = \frac{1}{\lambda} R^\mathsf{T} (\boldsymbol{x}_n - \overline{\boldsymbol{x}}) \tag{33.15}$$

with

- the weighted centroid

$$\overline{\boldsymbol{x}} = \frac{\sum_n w_n \boldsymbol{x}_n}{\sum_n w_n} \,. \tag{33.16}$$

- the rotation angle and the scale from the centred moment matrix

$$M = \sum_n w_n (\boldsymbol{x}_n - \overline{\boldsymbol{x}})(\boldsymbol{x}_n - \overline{\boldsymbol{x}})^\mathsf{T} \tag{33.17}$$

  via

$$\phi_0 = \frac{1}{2} \text{atan2}\left(-2M_{12}, M_{11} - M_{22}\right), \quad \lambda = \sqrt{\text{tr}(M)} \tag{33.18}$$

  leading to the rotation matrix

$$R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} . \tag{33.19}$$

The inverse transformation reads as

$$\boldsymbol{x}_n = \lambda R \boldsymbol{x}'_n + \overline{\boldsymbol{x}} \,. \tag{33.20}$$

From now on we assume the coordinates are conditioned following (33.15). The result will be finally back transformed using (33.20).

We now can derive the uncertainty of the estimated circle segment, see Fig. 33.1.

We represent it with the three parameters referring to the centroid coordinate system

$$s: \qquad \widehat{\boldsymbol{s}} = \begin{bmatrix} \widehat{h} \\ \widehat{\phi} \\ \widehat{\kappa} \end{bmatrix} \tag{33.21}$$

where h is the coordinate of the tangent point in the local coordinate system.

We derive it by from the circle parameters by

- in the centroid coordinate system

$$\boxed{\widehat{\boldsymbol{s}}'(\widehat{\boldsymbol{\theta}}') = \begin{bmatrix} \widehat{h}' \\ \widehat{\phi}' \\ \widehat{\kappa}' \end{bmatrix} = \begin{bmatrix} \widehat{x}'_0 + \widehat{r}' \\ \operatorname{atan2}(\widehat{y}'_0, \widehat{r}) \\ 1/\widehat{r} \end{bmatrix}} \tag{33.22}$$

  with the Jacobian

$$\boldsymbol{J}_{s'x'} = \frac{\partial \boldsymbol{s}'}{\partial \boldsymbol{x}'} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & \frac{r'}{y_0'^2+r'^2} & -\frac{y'_0}{y_0'^2+r'^2} \\ 0 & 0 & -\frac{1}{r'^2} \end{bmatrix}. \tag{33.23}$$

- in the original system

$$\widehat{\boldsymbol{s}}(\widehat{\boldsymbol{s}}') = \begin{bmatrix} \lambda \widehat{h}' \\ \widehat{\phi}' + \phi_0 \\ \frac{\widehat{r}}{\lambda} \end{bmatrix}, \quad \boldsymbol{J}_{ss'} = \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}'} = \begin{bmatrix} \lambda & & \\ & 1 & \\ & & \frac{1}{\lambda} \end{bmatrix}. \tag{33.24}$$

Thus, the covariance matrix of the circle element in the original coordinate system is

$$\boldsymbol{\Sigma}_{\widehat{s}\widehat{s}} = \boldsymbol{J}_{ss'}\, \boldsymbol{J}_{s'x'}\, \boldsymbol{\Sigma}_{\widehat{x}'\widehat{x}'}\, \boldsymbol{J}_{s'x'}^{\mathsf{T}}\, \boldsymbol{J}_{ss'}^{\mathsf{T}}. \tag{33.25}$$

In detail we obtain

1. Uncertainty of position of the tangent point across the circle, taking the correlations into account
$$\sigma_{\widehat{h}}^2 = \sigma_{\widehat{x}_0}^2 - 2\sigma_{\widehat{x}_0\widehat{r}} + \sigma_{\widehat{r}}^2. \tag{33.26}$$
2. Uncertainty of direction ϕ of tangent
$$\sigma_{\widehat{\phi}}^2 = \frac{1}{\widehat{y}_0'^2 + \widehat{r}'^2}(r'^2\sigma_{\widehat{y}'_0}^2 - 2\widehat{y}_0\widehat{r}'\sigma_{\widehat{y}_0\widehat{r}} + \widehat{y}_0^2\sigma_{r'}^2). \tag{33.27}$$
3. Uncertainty of curvature $\kappa = 1/r$
$$\sigma_{\widehat{\kappa}} = \frac{\sigma_{\widehat{r}}}{\widehat{r}^2}. \tag{33.28}$$
   Observe, the relative accuracy of the curvature and the radius are the same: $\sigma_{\widehat{r}}/\widehat{r} = \sigma_{\widehat{\kappa}}/\widehat{\kappa}$.
4. All elements are correlated. However, only the position ${}^t x_0$ and the radius show significant correlation.

The final representation for an uncertain circle segment is

$$s: \qquad \{x_t, y_t, \phi, \kappa; \mathbb{D}([q, \phi, \kappa])\}. \tag{33.29}$$

where (x_t, y_t, ϕ) defines the coordinate system of the tangent $\boldsymbol{t}$. The observation for $q = 0$, which therefore needs not stored, it is contained in the fixed coordinates (x_t, y_t). However, $\underline{q}$ is uncertain by σ_q. The complete likelihood function is

$$[\underline{q}; \underline{\phi}; \underline{\kappa}] \mid \mathcal{C} \sim \mathcal{M}\left([0; \phi; \kappa], \mathbb{D}([q; \phi; \kappa])\right) . \tag{33.30}$$

From this we easily can derive the uncertain tangent line $\underline{\mathbf{t}}$ from

$$\underline{\mathbf{t}} = \underline{T}^* \, \underline{\mathbf{t}}' \tag{33.31}$$

with

$$\underline{\mathbf{t}}' = \begin{bmatrix} 1 \\ \underline{\phi} \\ -\underline{q} \end{bmatrix} \tag{33.32}$$

and

$$\underline{T} = \begin{bmatrix} \cos\phi & -\sin\phi & x_t \\ \sin\phi & \cos\phi & y_t \\ 0 & 0 & 1 \end{bmatrix} \tag{33.33}$$

which represents two degrees of freedom, namely the shift q across the tangent and the direction ϕ of the normal. The third parameter is the uncertain curvature $\underline{\kappa}$.

34 Geometric Algebra and Projective Geometry for Representing Relations between Geometric Elements

Geometric relations and constructions of projective elements usually are represented with homogeneous vectors and matrices. We analyse these relations and constructions using Geometric or Clifford algebra. It confirms all representations for relating homogeneous representations of 3D points, planes and 3D lines, especially the sign choices.

34.1 Preface

This note (2003) is motivated by the very specific question, how to choose the sign when dualling the Plücker coordinates of a 3D line. It is based on the analysis of the geometric relations and constructions using Geometric or Clifford algebra. As an important side effect, it confirms all sign choices made for representing and relating homogeneous representations of 3D points, planes and 3D lines, finally documented in Förstner and Wrobel (2016, Chapt. 5), see the remark on p. 236.

Specifically, we originally (before 2000) chose to define the dual of a line with Plücker coordinates $\mathbf{L} = [\boldsymbol{L}_h, \boldsymbol{L}_0]$ as $\overline{\mathbf{L}}^\mathsf{T} = -[\boldsymbol{L}_0, \boldsymbol{L}_h]$, which was motivated by the cap product of the Grassmann Caley algebra Faugeras and Luong (2001, Def. 3.8, p.. 150), Faugeras and Papadopoulo (1998, meet-operator, p. 1125),and Browne (2009) and also follows from tensor notation for Plücker matrices $G_{mn} = \varepsilon_{klmn} G^{kl}$. It differs from our later work in Förstner et al. (2000) and in Geometric or Clifford Algebra where $\overline{\mathbf{L}}^\mathsf{T} = [\boldsymbol{L}_0, \boldsymbol{L}_h]$.

This note clarifies the reason, proposes to use the definition of the dual line in accordance with the Geometric Algebra, and shows that our construction results are not affected by this.

Moreover, all the relations/constructions are shown to be directly derivable from geometric algebra.

34.2 Relations between different representations for geometry

We collect representations and incidence relations for the following types of representation in 2D and 3D:

- Euclidean geometry
- Projective geometry
- Grassmann-Cayley algebra (GCA)
- Geometric or Clifford algebra (GA)

The representations are collected in the following Tab. 34.1, at the same time specifying the notation.

name	Entity	d.o.f.	Euclidean	projective	GCA	GA
$\mathcal{x}$	2d point	2	$\boldsymbol{x}$	$\mathbf{x}$	$\mathbf{x}$	$< x >_1$
$\mathcal{l}$	2d line	2	$(\boldsymbol{x}, \boldsymbol{d}), \lvert\boldsymbol{d}\rvert = 1$	$\mathbf{l}$	$\mathbf{l}$	$< l >_2$
$\mathcal{X}$	3d point	3	$\boldsymbol{X}$	$\mathbf{X}$	$\mathbf{X}$	$< X >_1$
$\mathcal{L}$	3d line	4	$(\boldsymbol{X}, \boldsymbol{D}), \lvert\boldsymbol{D}\rvert = 1$	$\mathbf{L}$	$\mathbf{L}$	$< L >_2$
$\mathcal{A}$	plane	3	$(\boldsymbol{N}, S), \lvert\boldsymbol{N}\rvert = 1$	$\mathbf{A}$	$\mathbf{A}$	$< A >_3$

Table 34.1: Representation of geometric entities. A line may be represented Euclideanly with a point and a direction, a plane may be represented Euclideanly by its normal and its distance to the origin. The indices for the elements in geometric algebra (GA) indicate what is called the grade of the vector, which can be visualized by the number of points necessary to generate it

Incidence relations are collected in the following Tab. tab:IncidenceOfGeometricEntities.

inc$(\boldsymbol{a}, \boldsymbol{b})$	d.o.f.	Euclidean	projective	GCA	GA
$\mathcal{x} \in \mathcal{l}$	1	$\boldsymbol{x} = \boldsymbol{x}_l + \lambda \boldsymbol{d}_l$	$\mathbf{x}^\mathsf{T}\mathbf{l} = 0$	$< \mathbf{x}, \mathbf{l} >= 0$	$x \wedge l = 0$
$\mathcal{X} \in \mathcal{A}$	1	$\boldsymbol{X}^\mathsf{T}\boldsymbol{N} = S$	$\mathbf{X}^\mathsf{T}\mathbf{A} = 0$	$< \mathbf{X}, \mathbf{A} >= 0$	$X \wedge A = 0$
$\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$	1	$\lvert\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{X}_2 - \boldsymbol{X}_1\rvert = 0$	$\mathbf{L}_1 D \mathbf{L}_2 = 0$	$< \mathbf{L}_1, \mathbf{L}_2 >= 0$	$L_1 \wedge L_2 = 0$
$\mathcal{X} \in \mathcal{L}$	2	$\boldsymbol{X} = \boldsymbol{X}_L + \lambda \boldsymbol{D}_L$	$\overline{\mathsf{\Gamma}}^\mathsf{T}(\mathbf{L})\mathbf{X} = \mathbf{0}$ $\overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{X})\mathbf{L} = \mathbf{0}$	$\mathbf{X} \wedge \mathbf{L} = \mathbf{0}$	$X \wedge L = 0$
$\mathcal{L} \in \mathcal{A}$	2	$\boldsymbol{L}_h \times \boldsymbol{N} = \mathbf{0}$ $\boldsymbol{X}_L^\mathsf{T}\boldsymbol{N} = S$	$\mathsf{\Gamma}(\mathbf{L})\mathbf{A} = \mathbf{0}$ $\mathsf{\Pi}^\mathsf{T}(\mathbf{A})\mathbf{L} = \mathbf{0}$	$\mathbf{L} \cap \mathbf{A} = \mathbf{0}$	$(A \wedge L)^* = 0$

Table 34.2: Representation of geometric incidences. The matrices $\mathsf{\Pi}(\cdot)$ and $\mathsf{\Gamma}(\cdot)$ are defined in Förstner and Wrobel (2016, (5.73),(7.37)) and given below, see (34.1) and (34.2)

.

34.3 Introduction to Geometric or Clifford Algebra

The description here is developed based on sources, namely Hestenes and Ziegler (1991), Browne (2009), Perwass (2000). The results partially have been realized with the MAPLE package `GA` by Mark Ashdown, see https://gitlab.com/majashdown/ga.

34.3.1 Basics

In Geometric Algebra (GA) all basic entities are collected in one. The basic entity is a vector.

Points are vectors

$$X = X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4 = \mathbf{X}^\mathsf{T}\boldsymbol{e}_X$$

with base vectors $\boldsymbol{e}_i, i = 1, 2, 3, 4$ collected in a compound base vector

$$\mathbf{e}_X = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \boldsymbol{e}_4 \end{bmatrix}$$

All further elements are built upon these base vectors as in Grassmann-Caley algebra (GCA).

The geometric product of two vectors is formally defined as

$$XY = X\cdot Y + X \wedge Y$$

It is a sum of the scalar product of X and Y, which is symmetric, thus $X\cdot Y = Y\cdot X$ and the outer product $X \wedge Y$ which is antisymmetric, thus $X \wedge Y = -Y \wedge X$. The sum of these both products is to be seen as the sum of two non-compatible elements such as 1 and i in complex numbers, e. g. $z = x + iy$, where there is no possibility to come to a sum of x and y.

Generally, GA works with multi-vectors, which are sums of r-vectors

$$A = \alpha + X + YZ + RST + EFGH + ...$$

If a multi-vector is an r-vector, it is homogeneous (not to be confused with our notion). The individual sums of the multi-vectors have grade 0, 1, 2, etc. We will only work with homogeneous vectors.

The geometric product of two vectors is called a bi-vector or a 2-vector. One also can multiply three of them XYZ, which yields a 3-vector.

For the base vectors $\boldsymbol{e}_i$ we have the following rules:

$$\boldsymbol{e}_i\cdot\boldsymbol{e}_j = \delta_{ij} \qquad \boldsymbol{e}_i \wedge \boldsymbol{e}_j = \begin{cases} 0, \text{ if } i = j \\ -\boldsymbol{e}_j \wedge \boldsymbol{e}_i, \text{ if } i \neq j \end{cases}$$

The left relation defines the metric in which we operate.

Thus, we have

$$\boldsymbol{e}_i\boldsymbol{e}_j = \begin{cases} 1, \text{ if } i = j \\ \boldsymbol{e}_i \wedge \boldsymbol{e}_j \quad \text{for } i \neq j \end{cases}$$

We often use the abbreviation

$$\boldsymbol{e}_i\boldsymbol{e}_j = \boldsymbol{e}_{ij}$$

Thus, there is no further simplification of the outer product of two different base vectors.

If we are in 4-dimensional space we have 4 base vectors, therefore 6 possible bi-vectors as basis

$$(\boldsymbol{e}_1 \wedge \boldsymbol{e}_2, \boldsymbol{e}_1 \wedge \boldsymbol{e}_3, \boldsymbol{e}_1 \wedge \boldsymbol{e}_4, \boldsymbol{e}_2 \wedge \boldsymbol{e}_3, \boldsymbol{e}_2 \wedge \boldsymbol{e}_4, \boldsymbol{e}_3 \wedge \boldsymbol{e}_4) = (\boldsymbol{e}_{12}, \boldsymbol{e}_{13}, \boldsymbol{e}_{14}, \boldsymbol{e}_{23}, \boldsymbol{e}_{24}, \boldsymbol{e}_{34})$$

If we look at triple products we have only four different ones, as in case two of the factors are identical we can reorder the factors and yield at least one factor 0. Therefore, we obtain the four base vectors:

$$(\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3, \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_4, \boldsymbol{e}_1\boldsymbol{e}_3\boldsymbol{e}_4, \boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_4) = (\boldsymbol{e}_{123}, \boldsymbol{e}_{124}, \boldsymbol{e}_{134}, \boldsymbol{e}_{234})$$

Finally, we look at quadruple products. There is only one:

$$\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_4 = \boldsymbol{e}_{1234}$$

If we now look at the squares of the base vectors we find:

1. base vectors for vectors
$$\boldsymbol{e}_i^2 = 1$$
2. base vectors for bi-vectors
$$\boldsymbol{e}_{ij}^2 = -1$$
as $\boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_i\boldsymbol{e}_j = -\boldsymbol{e}_i\boldsymbol{e}_j^2\boldsymbol{e}_i = -1$
3. base vectors for triple vectors
$$\boldsymbol{e}_{ijk}^2 = -1$$
as $\boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k\boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k = \boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k\boldsymbol{e}_k\boldsymbol{e}_i\boldsymbol{e}_j = -1$
4. base of quadtruple vectors
$$\boldsymbol{e}_{1234}^2 = 1$$

We need the reverse of a r-vector, which is just the product of its factors in the reverse order:
$$(XY...Z)^\dagger = Z...YX$$
We then can define the absolute value of an r-vector:
$$T = XY...Z \qquad |T|^2 = T^\dagger T$$
This enables us to define an inverse of an r-vector
$$T^{-1} = T^\dagger/|T|^2$$

Now observe
$$\boldsymbol{e}_{1234}\dagger = \boldsymbol{e}_{4321}$$
and
$$(\boldsymbol{e}_{1234})^{-1} = \boldsymbol{e}_{4321}/|\boldsymbol{e}_{1234}|^2 = \boldsymbol{e}_{1234}$$
In spite $\boldsymbol{e}_{1234}$ is not 1, it behaves like 1. Therefore, it is called the unit pseudo scalar
$$I = \boldsymbol{e}_{1234} \qquad \text{with} \qquad I^{-1} = I^\dagger = I$$

Remark: This specific property only holds, because we have used the special metric $\boldsymbol{e}_i\boldsymbol{e}_j = \delta_{ij}$. If we had taken a different metric, e. g. $\boldsymbol{e}_i^2 = 1, i = 1, 2, 3$ and $\boldsymbol{e}_4^2 = -1$ we had obtained $I^2 = -1$ and therefore $I^{-1} = -I$.

We now can define lines as bi-vectors
$$L = X \wedge Y$$
In case we use the basis
$$\boldsymbol{e}_L = [\boldsymbol{e}_4 \wedge \boldsymbol{e}_1, \boldsymbol{e}_4 \wedge \boldsymbol{e}_2, \boldsymbol{e}_4 \wedge \boldsymbol{e}_3, \boldsymbol{e}_2 \wedge \boldsymbol{e}_3, \boldsymbol{e}_3 \wedge \boldsymbol{e}_1, \boldsymbol{e}_1 \wedge \boldsymbol{e}_2]^\mathsf{T} = [\boldsymbol{e}_{41}, \boldsymbol{e}_{42}, \boldsymbol{e}_{43}, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}]^\mathsf{T}$$
we arrive at our definition of the Plücker coordinates of the line
$$L = \mathbf{L}^\mathsf{T}\boldsymbol{e}_L = L_1\boldsymbol{e}_{41} + L_2\boldsymbol{e}_{42} + L_3\boldsymbol{e}_{43} + L_4\boldsymbol{e}_{23} + L_5\boldsymbol{e}_{31} + L_6\boldsymbol{e}_{12}$$
Observe, the outer product of two vectors is a bi-vector, thus a homogeneous multi-vector of grade 2.

Remark: The correspondence of the coordinates L_i to the base vectors $\boldsymbol{e}_{kl}$ is arbitrary. We have chosen them, such that the first and second triple of $\boldsymbol{L}$ correspond to the direction and the moment pf the 3D line, the moment being the normal on the plane through L and the origin. $\diamond$

In a similar manner we can define planes
$$A = \mathbf{A}^\mathsf{T}\boldsymbol{e}_A$$
where we use the compound base vector
$$\mathbf{e}_A = \begin{bmatrix} \boldsymbol{e}_{234} \\ \boldsymbol{e}_{314} \\ \boldsymbol{e}_{124} \\ \boldsymbol{e}_{321} \end{bmatrix}$$
The reason for this choice results from the definition of duality.

34.3.2 Duality

The dual V^* of a multi-vector V is defined as

$$V^* = V\ I^{-1}$$

Thus we have the dual to a point

$$X^* = (X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4)\ \boldsymbol{e}_{1234} = X_1\boldsymbol{e}_{234} + X_2\boldsymbol{e}_{314} + X_3\boldsymbol{e}_{124} + X_4\boldsymbol{e}_{321}$$

This is a plane with identical coordinates, corresponding to the dualling operation in coordinate vectors:

$$\overline{\mathbf{X}} = I_4\mathbf{X}$$

Observe, the sequence of the first two indices in the first three base vectors (23,31,12).

Then we have the dual of a line

$$\begin{aligned} L^* &= (L_1\boldsymbol{e}_{41} + L_2\boldsymbol{e}_{42} + L_3\boldsymbol{e}_{43} + L_4\boldsymbol{e}_{23} + L_5\boldsymbol{e}_{31} + L_6\boldsymbol{e}_{12})\ \boldsymbol{e}_{1234} \\ &= L_1\boldsymbol{e}_{23} + L_2\boldsymbol{e}_{31} + L_3\boldsymbol{e}_{12} + L_4\boldsymbol{e}_{41} + L_5\boldsymbol{e}_{42} + L_6\boldsymbol{e}_{43} \\ &= L_4\boldsymbol{e}_{41} + L_5\boldsymbol{e}_{42} + L_6\boldsymbol{e}_{43} + L_1\boldsymbol{e}_{23} + L_2\boldsymbol{e}_{31} + L_3\boldsymbol{e}_{12} \end{aligned}$$

(as we had before last autumn! ... before 2003.)

Thus in coordinate vectors we have

$$\overline{\mathbf{L}} = D\mathbf{L}$$

with

$$D = \begin{bmatrix} \mathbf{0} & I_3 \\ I_3 & \mathbf{0} \end{bmatrix}$$

And finally, the dual of a plane

$$A^* = (A_1\boldsymbol{e}_{234} + A_2\boldsymbol{e}_{314} + A_3\boldsymbol{e}_{124} + A_4\boldsymbol{e}_{321})\ \boldsymbol{e}_{1234} = A_1\boldsymbol{e}_1 + A_2\boldsymbol{e}_2 + A_3\boldsymbol{e}_3 + A_4\boldsymbol{e}_4$$

yielding a point, in coordinates

$$\overline{\mathbf{A}} = I_4\mathbf{A}$$

34.3.3 The join

The join of two entities is the outer product in case they are not incident. In GA the outer product is the smallest subspace containing both elements, even in case the two elements have some space in common.

Thus, we have the following cases

- Line L as join of two distinct points

  $$L = X \wedge Y = -Y \wedge X$$

  The minus sign results from the anti-symmetry of the base-vectors $\boldsymbol{e}_i\boldsymbol{e}_j = -\boldsymbol{e}_j\boldsymbol{e}_i$
  The result is the following

  $$L = (X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4) \wedge (Y_1\boldsymbol{e}_1 + Y_2\boldsymbol{e}_2 + Y_3\boldsymbol{e}_3 + Y_4\boldsymbol{e}_4)$$

  which - after some manipulation - leads to

  $$L = \mathbf{L}^{\mathsf{T}}\boldsymbol{e}_L$$

  with

  $$\mathbf{L} = \Pi(\mathbf{X})\mathbf{Y} = -\Pi(\mathbf{Y})\mathbf{X}$$

  using our convention of $\Pi(\mathbf{X})$, cf. (34.1).
  Observe, we can interpret $X \wedge Y$ as a line only when inspecting the elements of $\mathbf{L}$, when seeing that $\mathbf{L}$ does not change when choosing two other point on the line as basis while noticing the intersection with a plane.

- Plane A as join a line L and a point X not sitting on L

$$A = L \wedge X = X \wedge L$$

The reason for the symmetry lies in the fact that $\boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k = \boldsymbol{e}_k\boldsymbol{e}_i\boldsymbol{e}_j$
The result is the following

$$A = (L_1\boldsymbol{e}_{41}+L_2\boldsymbol{e}_{42}+L_3\boldsymbol{e}_{43}+L_4\boldsymbol{e}_{23}+L_5\boldsymbol{e}_{31}+L_6\boldsymbol{e}_{12})\wedge(X_1\boldsymbol{e}_1+X_2\boldsymbol{e}_2+X_3\boldsymbol{e}_3+X_4\boldsymbol{e}_4)$$

which - after some manipulation - leads to

$$A = \mathbf{A}^{\mathsf{T}}\boldsymbol{e}_A$$

with

$$\mathbf{A} = \overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X})\mathbf{L} = \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathbf{L})\mathbf{X}$$

(cf. (34.3)) and

$$\overline{\mathsf{\Pi}}(\mathbf{X}) = D\,\mathsf{\Pi}(\mathbf{X})$$

and our Plücker-Matrix, cf. (34.2)

$$\overline{\mathsf{\Gamma}}(\mathbf{L}) = \mathsf{\Gamma}(\overline{\mathbf{L}})$$

- Then we have the join of a point X and a plane A not passing through the point

$$\beta I = X \wedge A = -A \wedge X$$

First, the minus sign results from $\boldsymbol{e}_{ijkl} = -\boldsymbol{e}_{jkli}$. Second, the result necessarily is a pseudo scalar, thus has unit $\boldsymbol{e}_{1234}$.
We explicitely get

$$\begin{aligned}\beta I &= (X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4) \wedge (A_1\boldsymbol{e}_{234} + A_2\boldsymbol{e}_{314} + A_3\boldsymbol{e}_{124} + A_4\boldsymbol{e}_{321}) \\ &= X_1A_1\boldsymbol{e}_1\boldsymbol{e}_{234} + X_2A_3\boldsymbol{e}_2\boldsymbol{e}_{314} + X_3A_3\boldsymbol{e}_3\boldsymbol{e}_{124} + X_4A_4\boldsymbol{e}_4\boldsymbol{e}_{321} = \mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{e}_{1234}\end{aligned}$$

The value β is the determinant of four points, in case A is the join of three points Y_i

$$\beta = |\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3| = -|\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \mathbf{X}|$$

- Finally, we have the outer product of two skew lines L and M

$$\gamma I = L \wedge M = M \wedge L$$

The symmetry results from $\boldsymbol{e}_{1234} = \boldsymbol{e}_{3412}$. Again the result must have unit $\boldsymbol{e}_{1234}$.
Explicitly we obtain

$$\begin{aligned}\gamma I &= (L_1\boldsymbol{e}_{41} + L_2\boldsymbol{e}_{42} + L_3\boldsymbol{e}_{43} + L_4\boldsymbol{e}_{23} + L_5\boldsymbol{e}_{31} + L_6\boldsymbol{e}_{12}) \\ &\quad \wedge(M_1\boldsymbol{e}_{41} + M_2\boldsymbol{e}_{42} + M_3\boldsymbol{e}_{43} + M_4\boldsymbol{e}_{23} + M_5\boldsymbol{e}_{31} + M_6\boldsymbol{e}_{12}) \\ &= L_1M_4\boldsymbol{e}_{41}\boldsymbol{e}_{23} + L_2M_5\boldsymbol{e}_{42}\boldsymbol{e}_{31} + ... \\ &= -L_1M_4\boldsymbol{e}_{1234} - L_2M_5\boldsymbol{e}_{1234} + ... \\ &= -\mathbf{L}^{\mathsf{T}}\overline{\mathbf{M}}\boldsymbol{e}_{1234}\end{aligned}$$

as all other products vanish, as they have at least one base vector $\boldsymbol{e}_i$ in common.
The scalar γ is the determinant of four points, in case the two lines are built by $L = X_1 \wedge X_2$ and $M = Y_1 \wedge Y_2$

$$\gamma = |\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2| = |\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{X}_1, \mathbf{X}_2|$$

All other products either are trivial, as they include a scalar as factor, or vanish, as they result in an entity with a product of more than 5 base vectors, which always vanish.

34.3.4 The meet

The meet of two multi-vectors in GA is formally defined by de Morgan's rule

$$(U \vee W)^* = U^* \wedge W^*$$

or explicitly

$$U \vee W = ((UI^{-1}) \wedge (WI^{-1}))I$$

Of course the meet is only defined in case the join of the two dual entities is defined. We have the following cases, dual-ling the previous four cases for the join

- Line L as the meet of two distinct planes

  $$L = A \vee B = -B \vee A = ((AI^{-1}) \wedge (BI^{-1})I$$

  This is in full agreement with our path of thinking.
  Therefore, we obtain

  $$L = ((A_1\boldsymbol{e}_1 + A_2\boldsymbol{e}_2 + A_3\boldsymbol{e}_3 + A_4\boldsymbol{e}_4) \wedge (B_1\boldsymbol{e}_1 + B_2\boldsymbol{e}_2 + B_3\boldsymbol{e}_3 + B_4\boldsymbol{e}_4))\ \boldsymbol{e}_{1234}$$

  and after some manipulation

  $$L = \mathbf{L}^{\mathsf{T}}\boldsymbol{e}_L$$

  with

  $$\mathbf{L} = \overline{\Pi}(\mathbf{A})\mathbf{B} = -\overline{\Pi}(\mathbf{B})\mathbf{A}$$

- Point X as the meet of a line L and a plane A not passing through the plane

  $$X = L \vee A = A \vee L = ((LI^{-1}) \wedge (AI^{-1}))I$$

  We obtain

  $$\begin{aligned} X \quad = \quad & ((L_1\boldsymbol{e}_{41} + L_2\boldsymbol{e}_{42} + L_3\boldsymbol{e}_{43} + L_4\boldsymbol{e}_{23} + L_5\boldsymbol{e}_{31} + L_6\boldsymbol{e}_{12})\boldsymbol{e}_{1234} \\ & \wedge(A_1\boldsymbol{e}_{234} + A_2\boldsymbol{e}_{314} + A_3\boldsymbol{e}_{124} + A_{321}\boldsymbol{e}_4)\boldsymbol{e}_{1234})\boldsymbol{e}_{1234} \end{aligned}$$

  which - after some manipulation yields

  $$X = \mathbf{X}^{\mathsf{T}}\boldsymbol{e}_X$$

  with

  $$\mathbf{X} = \Pi^{\mathsf{T}}(A)\mathbf{L} = \Gamma^{\mathsf{T}}(\mathbf{L})\mathbf{A}$$

- The meet of a point X and a plane A not passing through the point

  $$\beta = X \vee A = -A \vee X = ((XI^{-1}) \wedge (AI^{-1}))I$$

  The result is a scalar, as the term inside the outer brackets is a pseudo-scalar, which, when multiplied with I yields a scalar.
  The value of β is identical to the one derived from the join.

- The meet of two lines

  $$\gamma = L \vee M = M \vee L = ((LI^{-1}) \wedge (MI^{-1}))I$$

  Again the resultant scalar γ is the same as above.

34.3.5 The cap product

Faugeras' cap-product Faugeras and Luong (2001, Def. 3.8, p. 150) refers to two homogeneous multi-vectors (GCA only knows about homogeneous multi-vectors) V_r and V_{n-r} and yields a scalar

$$< V_r, V_{n-r} >_r = < V_{n-r}, V_r > = V_r \cdot V_{n-r} I^{-1} \stackrel{GA}{=} (V_r \wedge V_{n-r}) I$$

It obviously is symmetric and has specialization mentioned in Faugeras/Luong:

$$< X, A >_1 = < A, X >_1 = X \cdot A^* = \mathbf{X}^\mathsf{T}\mathbf{A} = \mathbf{A}^\mathsf{T}\mathbf{X}$$

and

$$< L, M >_2 = < L, M >_2 = L \cdot M^* = -\mathbf{L}^\mathsf{T}\mathbf{M}$$

There obviously is a third cap-product, not mentioned in Faugeras:

$$< A, X >_3 = < X, A >_3 = A \cdot X^* = -\mathbf{A}^\mathsf{T}\mathbf{X} = -\mathbf{X}^\mathsf{T}\mathbf{A}$$

The signs of the inner products result from the squares of the base vectors, $\boldsymbol{e}_i^2 = 1$ for $r = 1$, $\boldsymbol{e}_{ij}^2 = -1$ for $r = 2$, and $\boldsymbol{e}_{ijk}^2 = -1$ for $r = 3$.

Remark: The cap product of the lines can be evaluated as follows:

$$\begin{aligned}
(\mathbf{L}^\mathsf{T}\boldsymbol{e}_L) \cdot (\mathbf{M}^\mathsf{T}\boldsymbol{e}_L^*) &= (\mathbf{L}^\mathsf{T}\boldsymbol{e}_L) \cdot (\mathbf{M}^\mathsf{T} D \boldsymbol{e}_L) \\
&= \sum_{i=1}^{6} \left(L_i \boldsymbol{e}_{L_i} \, (\sum_{j=1}^{6} D_{ij} M_j) \boldsymbol{e}_{L_i} \right) \\
&= -\mathbf{L}^\mathsf{T} D \mathbf{M}
\end{aligned}$$

as

$$\boldsymbol{e}_{L_i}^2 = -1 \qquad \text{for all } i$$

◇

34.3.6 Influence onto Definition of Dual Line and the Relations

The reasoning for the dual-ling of the 3D-line needs to be reconsidered

Thus, there are two processes to be distinguished:

1. dual-ling, here exchanging homogeneous and Euclidean part
2. calculating the cap-product, resulting in a sign change, due to the metric induced by the base vectors of the line.

So, the argument for taking the negative exchanged subvectors as dual-ling operation was misled by the sign change induced by the - not-visible - base vectors.

Therefore, I recommend to go back, and have the original dual-ling operation for 3D-lines, which is simpler and consistent with the more general rules of GA.

Moreover, nothing changes with the relations. All tables stay the same.

34.4 GCA as Specialization of the GA

As mentioned GA is a generalization of Grassmann-Caley algebra.

We may interpret 1-vectors are point, 2-vectors are 3D-lines, and 3-vectors are planes.

Together with scalars and pseudo-scalars we have a multi-vector

$$A = \alpha + \mathbf{X}^\mathsf{T}\mathbf{e}_X + \mathbf{L}^\mathsf{T}\mathbf{e}_L + \mathbf{A}^\mathsf{T}\mathbf{e}_A + \beta I$$

with the compound base vectors for points

$$\mathbf{e}_X = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \boldsymbol{e}_4 \end{bmatrix}$$

the compound base vectors for lines

$$\mathbf{e}_L = \begin{bmatrix} \boldsymbol{e}_{41} \\ \boldsymbol{e}_{42} \\ \boldsymbol{e}_{43} \\ \boldsymbol{e}_{23} \\ \boldsymbol{e}_{31} \\ \boldsymbol{e}_{12} \end{bmatrix} \qquad \text{with} \qquad \boldsymbol{e}_{ij} = \boldsymbol{e}_i \wedge \boldsymbol{e}_j = -\boldsymbol{e}_{ji}$$

the compound base vectors for planes

$$\mathbf{e}_A = \begin{bmatrix} \boldsymbol{e}_{234} \\ \boldsymbol{e}_{314} \\ \boldsymbol{e}_{124} \\ \boldsymbol{e}_{321} \end{bmatrix}$$

and the unit pseudo scalar

$$I = \boldsymbol{e}_{1234} = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4$$

Dualling an entity is achieved by post multiplication with

$$I^{-1} = \boldsymbol{e}_{4321} = \boldsymbol{e}_4 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_1$$

which in this special case has the property

$$I = I^{-1}$$

Thus, we obtain

$$A^* = A\, I^{-1}$$

If we now dualize the base vectors for points, lines and planes we obtain using $\boldsymbol{e}_i^2 = 1$ (there was an error in the previous version)

$$\mathbf{e}_X^* = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \boldsymbol{e}_4 \end{bmatrix} I^{-1} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \boldsymbol{e}_4 \end{bmatrix} \wedge \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4 = \begin{bmatrix} \boldsymbol{e}_{234} \\ \boldsymbol{e}_{314} \\ \boldsymbol{e}_{124} \\ \boldsymbol{e}_{321} \end{bmatrix} = \mathbf{e}_A$$

and therefore

$$\mathbf{e}_A^* = \mathbf{e}_X$$

But we obtain

$$\mathbf{e}_L^* = \begin{bmatrix} \boldsymbol{e}_{41} \\ \boldsymbol{e}_{42} \\ \boldsymbol{e}_{43} \\ \boldsymbol{e}_{23} \\ \boldsymbol{e}_{31} \\ \boldsymbol{e}_{12} \end{bmatrix} \wedge \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4 = \begin{bmatrix} \boldsymbol{e}_{23} \\ \boldsymbol{e}_{31} \\ \boldsymbol{e}_{12} \\ \boldsymbol{e}_{41} \\ \boldsymbol{e}_{42} \\ \boldsymbol{e}_{43} \end{bmatrix}$$

34.5 Statistical error propagation

Given a nonlinear function $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}})$ of stochastical variables, which are underlined, the first order approximation of error propagation leads to the following result:

Given $\underline{\boldsymbol{x}} \sim M(\boldsymbol{\mu}_x, \Sigma_{xx})$ represented with its first and second moments $\boldsymbol{\mu}_x$ and Σ_{xx} then the distribution of $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}})$ is

$$\underline{\boldsymbol{y}} \sim M(\boldsymbol{f}(\boldsymbol{\mu}_x, J\Sigma_{xx}J^\mathsf{T})$$

with the Jacobian

$$J = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

Bilinear forms

$$\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b} \qquad \text{or} \qquad c_k = \sum_{ij} f_{ijk} a_i b_j$$

can be written as

$$\boldsymbol{c} = U(\boldsymbol{b})\boldsymbol{a} = V(\boldsymbol{a})\boldsymbol{b}$$

with

$$U_{ki} = \sum_j f_{ijk} b_j \qquad V_{kj} = \sum_i f_{ijk} a_i$$

or

$$U = \frac{\partial(\boldsymbol{a} \circ \boldsymbol{b})}{\partial \boldsymbol{a}} \qquad V = \frac{\partial(\boldsymbol{a} \circ \boldsymbol{b})}{\partial \boldsymbol{b}}$$

Since all the constructions discussed above a bi-linear or multi-linear, we easily may perform variance propagation based on the coordinates of the entities. Care has to be taken, since the degrees of freedom usually is less than the number of coordinates, thus the coordinates of the homogeneous entities usually have singular covariance matrices.

34.6 Geometric algebra with coordinate vectors

The representation in geometric algebra (GA) is with vectors represented as linear forms of base vectors.

Applied to projective geometry (PG) it is useful to use G_4. Then we have the following relations between elements in GA and PG, including scalars and results of 4×4-determinants

1. Scalars. Any scalar α can be interpreted as a "multi-vector"[1] $< D >_0$ of grade 0

$$\alpha =< D >_0 \qquad \leftrightarrow \qquad \alpha$$

2. 3D points vectors

$$X =< D >_1 = X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4 \qquad \leftrightarrow \qquad \mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

3. 3D lines are bi-vectors

$$L =< D >_2 = L_1\boldsymbol{e}_{4,1} + L_2\boldsymbol{e}_{4,2} + L_3\boldsymbol{e}_{4,3} + L_4\boldsymbol{e}_{2,3} + L_5\boldsymbol{e}_{3,1} + L_6\boldsymbol{e}_{1,2} \qquad \leftrightarrow \qquad \mathbf{L} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{bmatrix}$$

with

$$\boldsymbol{e}_{ij} = \boldsymbol{e}_i\boldsymbol{e}_j$$

[1] In the original note the indices had an overline.

4. Planes are tri-vectors

$$A =< D >_3 = A_1\boldsymbol{e}_{2,3,4} + A_2\boldsymbol{e}_{3,1,4} + A_3\boldsymbol{e}_{1,2,4} + A_4\boldsymbol{e}_{3,2,1} \qquad \leftrightarrow \qquad \mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}$$

$$\boldsymbol{e}_{ijk} = \boldsymbol{e}_i\boldsymbol{e}_j\boldsymbol{e}_k$$

5. Pseudoscalar being the result of 4×4 determinants

$$\beta I =< D >_4 = \beta\boldsymbol{e}_{1,2,3,4} \qquad \leftrightarrow \qquad \beta$$

In PG one can write the join of two elements, e. g. points as

$$\mathbf{L} = \mathbf{X} \wedge \mathbf{Y} = \mathsf{\Pi}(\mathbf{X})\mathbf{Y} = -\mathsf{\Pi}(\mathbf{Y})\mathbf{X}$$

with

$$\mathsf{\Pi}(\mathbf{X}) = \frac{\partial(\mathbf{X} \wedge \mathbf{Y})}{\partial \mathbf{Y}} = \left(\begin{array}{ccc|c} T & 0 & 0 & -U \\ 0 & T & 0 & -V \\ 0 & 0 & T & -W \\ \hline 0 & -W & V & 0 \\ W & 0 & -U & 0 \\ -V & U & 0 & 0 \end{array}\right) \tag{34.1}$$

Also we have the join of a line $\mathbf{L}$ and a point $\mathbf{X}$

$$\mathbf{A} = \mathbf{L} \wedge \mathbf{X} = \mathbf{X} \wedge \mathbf{L} = \mathbf{A} = \overline{\mathsf{\Gamma}}^{\mathsf{T}}(\mathbf{L})\mathbf{X} = \overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X})\mathbf{L}$$

with the Plückermatrix

$$\mathsf{\Gamma}(\mathbf{L}) = \left(\begin{array}{ccc|c} 0 & L_6 & -L_5 & -L_1 \\ -L_6 & 0 & L_4 & -L_2 \\ L_5 & -L_4 & 0 & -L_3 \\ \hline L_1 & L_2 & L_3 & 0 \end{array}\right) = -\mathsf{\Gamma}^{\mathsf{T}}(\mathbf{L}) \tag{34.2}$$

the dual line (**new version !!!**)

$$\overline{\mathbf{L}} = D\mathbf{L} \qquad D = \begin{bmatrix} & I_3 \\ I_3 & \end{bmatrix}$$

and the matrices

$$\overline{\mathsf{\Pi}}(\mathbf{X}) \doteq D\mathsf{\Pi}(\mathbf{X}) \qquad \overline{\mathsf{\Gamma}}(\mathbf{L}) = \mathsf{\Gamma}(\overline{\mathbf{L}}) = \left(\begin{array}{ccc|c} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ \hline L_4 & L_5 & L_6 & 0 \end{array}\right) \tag{34.3}$$

In order to be able to apply statistical error propagation we need to represent a multivector as vector. Then, as the geometric product is bilinear, we can directly arrive at the Jacobian of a geometric product with respect to its factors.

We apply the general mapping in G_N

$$D = \sum_{n=1}^{N} < D >_{\overline{n}} \qquad \rightarrow \qquad \underset{2^N \times 1}{\boldsymbol{d}} = \begin{bmatrix} < D >_{\overline{0}} \\ .. \\ < D >_{\overline{n}} \\ ... \\ < D >_{\overline{N}} \end{bmatrix}$$

In the case of PG we have the mapping

$$D \qquad \rightarrow \underset{16\times 1}{\boldsymbol{d}} = \begin{bmatrix} \alpha \\ \mathbf{X} \\ \mathbf{L} \\ \mathbf{A} \\ \beta \end{bmatrix}$$

Therefore, we can attach uncertainty information to a multi-vector by working with the pair

$$(\boldsymbol{d}, \Sigma_{dd})$$

in the case of PG requiring a 16×16-covariance matrix for every multi-vector, even in case is is a homogeneous one.

The overhead in representation is the price for having a compact notation. However, one always can represent all entities with sparse techniques, equivalent to only storing the coefficients of the base-vectors which are non-zero.

The idea can be transferred to any geometric algebra with bilinear forms involved.

In the following we give the Jacobians for the geometric product $C_g = A_g B_g$, the join $C_g = A_g \wedge B_g$ and the meet $C_g = A_g \vee B_g = A_g \cap B_g = (A_g I^{-1} \wedge B_g I^{-1})I$ for the three following multi-vectors

$$\begin{aligned} A_g \;=\; & \alpha + \\ & + X_1\boldsymbol{e}_1 + X_2\boldsymbol{e}_2 + X_3\boldsymbol{e}_3 + X_4\boldsymbol{e}_4 + \\ & L_1\boldsymbol{e}_{4,1} + L_2\boldsymbol{e}_{4,2} + L_3\boldsymbol{e}_{4,3} + L_4\boldsymbol{e}_{2,3} + L_5\boldsymbol{e}_{3,1} + L_6\boldsymbol{e}_{1,2} \\ & + A_1\boldsymbol{e}_{2,3,4} + A_2\boldsymbol{e}_{3,1,4} + A_3\boldsymbol{e}_{1,2,4} + A_4\boldsymbol{e}_{3,2,1} \\ & + \beta I \end{aligned}$$

$$\begin{aligned} B_g \;=\; & \gamma + \\ & + Y_1\boldsymbol{e}_1 + Y_2\boldsymbol{e}_2 + Y_3\boldsymbol{e}_3 + Y_4\boldsymbol{e}_4 + \\ & M_1\boldsymbol{e}_{4,1} + M_2\boldsymbol{e}_{4,2} + M_3\boldsymbol{e}_{4,3} + M_4\boldsymbol{e}_{2,3} + M_5\boldsymbol{e}_{3,1} + M_6\boldsymbol{e}_{1,2} \\ & + B_1\boldsymbol{e}_{2,3,4} + B_2\boldsymbol{e}_{3,1,4} + B_3\boldsymbol{e}_{1,2,4} + B_4\boldsymbol{e}_{3,2,1} \\ & + \delta I \end{aligned}$$

and

$$\begin{aligned} C_g \;=\; & \varepsilon + \\ & + Z_1\boldsymbol{e}_1 + Z_2\boldsymbol{e}_2 + Z_3\boldsymbol{e}_3 + Z_4\boldsymbol{e}_4 + \\ & N_1\boldsymbol{e}_{4,1} + N_2\boldsymbol{e}_{4,2} + N_3\boldsymbol{e}_{4,3} + N_4\boldsymbol{e}_{2,3} + N_5\boldsymbol{e}_{3,1} + N_6\boldsymbol{e}_{1,2} \\ & + C_1\boldsymbol{e}_{2,3,4} + C_2\boldsymbol{e}_{3,1,4} + C_3\boldsymbol{e}_{1,2,4} + C_4\boldsymbol{e}_{3,2,1} \\ & + \eta I \end{aligned}$$

represented as vectors

$$\boldsymbol{a} = \begin{bmatrix} \alpha \\ \mathbf{X} \\ \mathbf{L} \\ \mathbf{A} \\ \beta \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta \end{bmatrix} \qquad \boldsymbol{c} = \begin{bmatrix} \varepsilon \\ \mathbf{Z} \\ \mathbf{N} \\ \mathbf{C} \\ \eta \end{bmatrix}$$

Thus, thin letters representing the geometric elements, the pseudo scalars including their unit

$$A_g = \alpha + X + L + A + \beta$$
$$B_g = \gamma + Y + M + B + \delta$$

$$C_g = \epsilon + Z + N + C + \eta$$

The following Jacobians are calculated with the Maple package "GA Package, Version 1.1, for Maple V Release 5", written by Mark Ashdown, Astrophysics Group, Cavendish Laboratory, University of Cambridge, see the enclosed Maple code.

Join of two elements

$$J_{a\wedge b} = \left(\frac{\partial \boldsymbol{a}\wedge \boldsymbol{b}}{\partial \boldsymbol{b}}\right)$$

$$\left[\begin{array}{c|cccc|cccccc|cccc|c}
\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
X_1 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X_2 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X_3 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X_4 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
L_1 & X_4 & 0 & 0 & -X_1 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L_2 & 0 & X_4 & 0 & -X_2 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L_3 & 0 & 0 & X_4 & -X_3 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L_4 & 0 & -X_3 & X_2 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L_5 & X_3 & 0 & -X_1 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 \\
L_6 & -X_2 & X_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\
\hline
A_1 & 0 & -L_3 & L_2 & L_4 & 0 & X_3 & -X_2 & X_4 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 \\
A_2 & L_3 & 0 & -L_1 & L_5 & -X_3 & 0 & X_1 & 0 & X_4 & 0 & 0 & \alpha & 0 & 0 & 0 \\
A_3 & -L_2 & L_1 & 0 & L_6 & X_2 & -X_1 & 0 & 0 & 0 & X_4 & 0 & 0 & \alpha & 0 & 0 \\
A_4 & -L_4 & -L_5 & -L_6 & 0 & 0 & 0 & 0 & -X_1 & -X_2 & -X_3 & 0 & 0 & 0 & \alpha & 0 \\
\hline
\beta & -A_1 & -A_2 & -A_3 & -A_4 & -L_4 & -L_5 & -L_6 & -L_1 & -L_2 & -L_3 & X_1 & X_2 & X_3 & X_4 & \alpha
\end{array}\right]$$

or

$$\left[\begin{array}{c} \varepsilon \\ \mathbf{Z} \\ \mathbf{N} \\ \mathbf{C} \\ \eta \end{array}\right] = \left[\begin{array}{ccccc} \alpha & & & & \\ \mathbf{X} & \alpha I & & & \\ \mathbf{L} & \overline{\Pi}(\mathbf{X}) & \alpha I & & \\ \mathbf{A} & \overline{\Gamma}^{\mathsf{T}}(\mathbf{L}) & \overline{\Pi}^{\mathsf{T}}(\mathbf{X}) & \alpha I & \\ \beta & -\mathbf{A}^{\mathsf{T}} & -\overline{\mathbf{L}}^{\mathsf{T}} & \mathbf{X}^{\mathsf{T}} & \alpha \end{array}\right] \left[\begin{array}{c} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta \end{array}\right]$$

Meet of two elements

$$J_{a \vee b} = \left(\frac{\partial \boldsymbol{a} \vee \boldsymbol{b}}{\partial \boldsymbol{b}} \right)$$

$$\left[\begin{array}{c|cccc|cccccc|cccc|c}
\beta & A_1 & A_2 & A_3 & A_4 & -L_4 & -L_5 & -L_6 & -L_1 & -L_2 & -L_3 & -X_1 & -X_2 & -X_3 & -X_4 & \alpha \\
\hline
0 & \beta & 0 & 0 & 0 & A_4 & 0 & 0 & 0 & A_3 & -A_2 & 0 & -L_6 & L_5 & L_1 & X_1 \\
0 & 0 & \beta & 0 & 0 & 0 & A_4 & 0 & -A_3 & 0 & A_1 & L_6 & 0 & -L_4 & L_2 & X_2 \\
0 & 0 & 0 & \beta & 0 & 0 & 0 & A_4 & A_2 & -A_1 & 0 & -L_5 & L_4 & 0 & L_3 & X_3 \\
0 & 0 & 0 & 0 & \beta & -A_1 & -A_2 & -A_3 & 0 & 0 & 0 & -L_1 & -L_2 & -L_3 & 0 & X_4 \\
\hline
0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & -A_3 & A_2 & 0 & L_1 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & A_3 & 0 & -A_1 & 0 & L_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & -A_2 & A_1 & 0 & 0 & L_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & A_4 & 0 & 0 & -A_1 & L_4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & A_4 & 0 & -A_2 & L_5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & A_4 & -A_3 & L_6 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & A_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & A_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & A_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & A_4 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta
\end{array}\right]$$

or

$$\begin{bmatrix} \varepsilon \\ \mathbf{Z} \\ \mathbf{N} \\ \mathbf{C} \\ \eta \end{bmatrix} = \begin{bmatrix} \beta & \mathbf{A}^{\mathsf{T}} & -\overline{\mathbf{L}}^{\mathsf{T}} & -\mathbf{X}^{\mathsf{T}} & \alpha \\ & \beta I & \Pi^{\mathsf{T}}(\mathbf{A}) & \Gamma^{\mathsf{T}}(\mathbf{L}) & \mathbf{X} \\ & & \beta I & \overline{\Pi}(\mathbf{A}) & \mathbf{L} \\ & & & \beta I & \mathbf{A} \\ & & & & \beta \end{bmatrix} \begin{bmatrix} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta \end{bmatrix}$$

Finally the dual

$$J_{b^*b} = \left(\frac{\partial \boldsymbol{b}^*}{\partial \boldsymbol{b}}\right)$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

or

$$\underbrace{\begin{bmatrix} \overline{\gamma} \\ \overline{\mathbf{Y}} \\ \overline{\mathbf{M}} \\ \overline{\mathbf{B}} \\ \overline{\delta} \end{bmatrix}}_{\boldsymbol{c}} = \underbrace{\begin{bmatrix} & & & & 1 \\ & & & I_4 & \\ & & D & & \\ & I_4 & & & \\ 1 & & & & \end{bmatrix}}_{V_{b*}(b)} \underbrace{\begin{bmatrix} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta \end{bmatrix}}_{\boldsymbol{b}}$$

As a comparison we obtain for the join

$$C_g = A_g \wedge B_g \qquad \underbrace{\begin{bmatrix} \varepsilon \\ \mathbf{Z} \\ \mathbf{N} \\ \mathbf{C} \\ \eta \end{bmatrix}}_{\boldsymbol{c}} = \underbrace{\begin{bmatrix} \alpha & & & & \\ \mathbf{X} & \alpha I & & & \\ \mathbf{L} & \overline{\Pi}(\mathbf{X}) & \alpha I & & \\ \mathbf{A} & \overline{\Gamma}^{\mathsf{T}}(\mathbf{L}) & \overline{\Pi}^{\mathsf{T}}(\mathbf{X}) & \alpha I & \\ \beta & -\mathbf{A}^{\mathsf{T}} & -\overline{\mathbf{L}}^{\mathsf{T}} & \mathbf{X}^{\mathsf{T}} & \alpha \end{bmatrix}}_{V_{a \wedge b}(\boldsymbol{a})} \underbrace{\begin{bmatrix} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta \end{bmatrix}}_{\boldsymbol{b}}$$

$$= \begin{bmatrix} \alpha \wedge \gamma \\ X \wedge \gamma + \alpha \wedge X \\ L \wedge \gamma + X \wedge Y + \alpha \wedge M \\ A \wedge \gamma + L \wedge Y + X \wedge M + \alpha \wedge B \\ \beta \wedge \gamma + A \wedge Y + L \wedge M + X \wedge B + \alpha \wedge \delta \end{bmatrix}$$

and for the meet

$$C_g = A_g \vee B_g = A_g \cap B_g \qquad \underbrace{\begin{bmatrix} \varepsilon \\ \mathbf{Z} \\ \mathbf{N} \\ \mathbf{C} \\ \eta I \end{bmatrix}}_{\boldsymbol{c}} = \underbrace{\begin{bmatrix} \beta & \mathbf{A}^{\mathsf{T}} & -\overline{\mathbf{L}}^{\mathsf{T}} & -\mathbf{X}^{\mathsf{T}} & \alpha I \\ & \beta I & \Pi^{\mathsf{T}}(\mathbf{A}) & \Gamma^{\mathsf{T}}(\mathbf{L}) & \mathbf{X} \\ & & \beta I & \overline{\Pi}(\mathbf{A}) & \mathbf{L} \\ & & & \beta I & \mathbf{A} \\ & & & & \beta \end{bmatrix}}_{\boldsymbol{V}_{a \vee b}(\boldsymbol{a})} \underbrace{\begin{bmatrix} \gamma \\ \mathbf{Y} \\ \mathbf{M} \\ \mathbf{B} \\ \delta I \end{bmatrix}}_{\boldsymbol{b}}$$

$$= \begin{bmatrix} \beta \cap \gamma + A \cap Y + L \cap M + X \cap B + \alpha \cap \delta \\ \beta \cap Y + A \cap L + L \cap B + X \cap \delta \\ \beta \cap M + A \cap B + L \cap \delta \\ \beta \cap B + A \cap \delta \\ \beta \cap \delta \end{bmatrix}$$

34.7 Conclusions

We may conclude the following:

1. The dualling of the line in GA is identical to exchanging the two 3-subvectors, which is simpler than taking the minus sign. This causes no irritation.
2. The consequences for the construction tables are negligible. Only the minus sign in the relation
$$< \mathbf{L}, \mathbf{M} >= -\mathbf{L}^{\mathsf{T}}\overline{\mathbf{M}} = |\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}|$$
needs to be taken into account.
3. The GA proves nicely our equations including all signs.
4. There is no need - at the moment - to switch to GA, which for our application would lead to an overhead.

35 Euler Angles and Small Multiplicative Rotation Vector

We provide the Jacobian for two representations of uncertain 3D rotations: the Euler angles and the multiplicative representation of an uncertain rotation. The Jacobian is singular, in case the second rotation angle is 90 degrees, which in the chosen Euler representation corresponds to the Gimbal lock.

35.1 Preface

The note (2019) provides the Jacobian for two representations of uncertain 3D rotations: the Euler angles and the multiplicative representation of an uncertain rotation. The Jacobian is singular, in case the second rotation angle is $\pm 90°$, which in the chose Euler representation corresponds to the Gimbal lock.

35.2 The Problem

A rotation can be represented by Euler angles with the vector

$$\boldsymbol{\theta} = \begin{bmatrix} \omega \\ \phi \\ \kappa \end{bmatrix} \tag{35.1}$$

e.g., as

$$R(\boldsymbol{\theta}) = R_3(\kappa) R_2(\phi) R_1(\omega) \tag{35.2}$$

and by a multiplicative representation with a small vector

$$\boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{35.3}$$

as

$$R(\boldsymbol{r}, R^a) = R(\boldsymbol{r}) R^a \,. \tag{35.4}$$

The task is to derive the Jacobian

$$J_{r\theta} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\theta}} \,. \tag{35.5}$$

35.3 Derivation

We start from the identity of the total derivative

$$\mathrm{d}R = \mathrm{d}R(\boldsymbol{\theta}) = \mathrm{d}R(\boldsymbol{r}, R^a)\,. \tag{35.6}$$

and aim at finding a relation between $\mathrm{d}\boldsymbol{\theta}$ and $\mathrm{d}\boldsymbol{r}$ under the assumption $R = R^a$, i.e., differential $\mathrm{d}r$.

We first obtain

$$\begin{aligned}
\mathrm{d}R(\boldsymbol{\theta}) &= \mathrm{d}\left(R_3(\kappa)R_2(\phi)R_1(\omega)\right) && (35.7)\\
&= \mathrm{d}R_3(\kappa)\,\left(R_2(\phi)R_1(\omega)\right) + R_3(\kappa)\,\mathrm{d}R_2(\phi)\,R_1(\omega) + \left(R_3(\kappa)R_2(\phi)\right)\mathrm{d}R_1(\omega) && (35.8)
\end{aligned}$$

Now we observe, e.g., for ω

$$\begin{aligned}
\mathrm{d}R_1(\omega) &= \mathrm{d}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} && (35.9)\\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin\omega & -\cos\omega \\ 0 & \cos\omega & -\sin\omega \end{bmatrix}\mathrm{d}\omega && (35.10)\\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & +1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} && (35.11)\\
&= S(\boldsymbol{e}_1)R_1(\omega)\,, && (35.12)
\end{aligned}$$

or generally

$$\mathrm{d}R_i(\alpha) = S(\boldsymbol{e}_i)R_i(\alpha)\,. \tag{35.13}$$

Similarly we thus have

$$\mathrm{d}R_2(\phi) = S(\boldsymbol{e}_2)R_2(\phi)\,\mathrm{d}\phi \quad \text{and} \quad \mathrm{d}R_3(\kappa) = S(\boldsymbol{e}_3)R_3(\kappa)\,\mathrm{d}\kappa \tag{35.14}$$

This leads to

$$\begin{aligned}
\mathrm{d}R(\boldsymbol{\theta}) &= S(\boldsymbol{e}_3)\,R_3(\kappa)\,R_2(\phi)\,R_1(\omega)\,\mathrm{d}\kappa + && (35.15)\\
&\quad R_3(\kappa)\,S(\boldsymbol{e}_2)\,R_2(\phi)\,R_1(\omega)\,\mathrm{d}\phi + && (35.16)\\
&\quad R_3(\kappa)\,R_2(\phi)\,S(\boldsymbol{e}_1)\,R_1(\omega)\,\mathrm{d}\omega && (35.17)
\end{aligned}$$

We now use the relation $R(\boldsymbol{a}\times\boldsymbol{b}) = R\boldsymbol{a}\times R\boldsymbol{b}$ which is valid for all $\boldsymbol{b}$ in the form

$$RS(\boldsymbol{a}) = S(R\boldsymbol{a})R \quad \text{or} \quad RS(\boldsymbol{a})R^{\mathsf{T}} = S(R\boldsymbol{a})\,. \tag{35.18}$$

Then we obtain

$$\begin{aligned}
\mathrm{d}R(\boldsymbol{\theta}) &= S(\boldsymbol{e}_3)R\,\mathrm{d}\kappa + && (35.19)\\
&\quad S(R_3(\kappa)\boldsymbol{e}_2)R\,\mathrm{d}\phi + && (35.20)\\
&\quad S(R_3(\kappa)R_2(\phi)\boldsymbol{e}_1)R\,\mathrm{d}\omega && (35.21)
\end{aligned}$$

or the skew symmetric matrix

$$\begin{aligned}
\mathrm{d}R(\boldsymbol{\theta})R^{\mathsf{T}} &= S(\boldsymbol{e}_3\mathrm{d}\kappa) + && (35.22)\\
&\quad S(R_3(\kappa)\boldsymbol{e}_2\mathrm{d}\phi) + && (35.23)\\
&\quad S(R_3(\kappa)R_2(\phi)\boldsymbol{e}_1\mathrm{d}\omega) && (35.24)
\end{aligned}$$

Now the total differential of $R(\boldsymbol{r}; R^a)$ is given by

$$\mathrm{d}R(\boldsymbol{r}; R^a) = S(\mathrm{d}\boldsymbol{r})R^a \tag{35.25}$$

Hence we have

$$\mathrm{d}R(\boldsymbol{r};R^a)R^{a\mathsf{T}} = S(\mathrm{d}\boldsymbol{r}) \tag{35.26}$$

Since the approximate rotation matrix is the point of linearization, we have the constraint

$$\mathrm{d}R(\boldsymbol{\theta})R^{\mathsf{T}} = \mathrm{d}R(\boldsymbol{r};R^a)R^{\mathsf{T}} \tag{35.27}$$

Therefore the two skew symmetric matrices (35.22) and (35.26) need to be identical. From this we follow

$$\boldsymbol{e}_3\mathrm{d}\kappa + R_3(\kappa)\boldsymbol{e}_2\mathrm{d}\phi + R_3(\kappa)R_2(\phi)\boldsymbol{e}_1\mathrm{d}\omega = \mathrm{d}\boldsymbol{r} \tag{35.28}$$

or

$$\boldsymbol{r} = J_{r\theta}\boldsymbol{\theta} \tag{35.29}$$

with the Jacobian

$$\boxed{J_{r\theta} = [R_3(\kappa)R_2(\phi)\boldsymbol{e}_1 \mid R_3(\kappa)\boldsymbol{e}_2 \mid \boldsymbol{e}_3]} \tag{35.30}$$

The determinant of the Jacobian is

$$|J_{r\theta}| = \cos\phi\,. \tag{35.31}$$

This is why for $\cos\phi = 0$ or for $\phi = \pm 90°$ there is no unique relation between $\mathrm{d}\boldsymbol{r}$ and $\mathrm{d}\boldsymbol{\theta}$, which is known as the gimbal lock. Observe, the rotation matrix (35.2) for $\phi = 90°$ specializes to

$$R(\alpha,\beta=90°,\gamma) = \begin{bmatrix} 0 & -\sin(\gamma-\alpha) & \cos(\gamma-\alpha) \\ 0 & \cos(\gamma-\alpha) & \sin(\gamma-\alpha) \\ -1 & 0 & 0 \end{bmatrix}, \tag{35.32}$$

indicating, that it only depends on the difference $\gamma - \alpha$ of two of the angles, i.e., loosing one degree of freedom.

36 Visualization of the Projective Space using Stereographic Projection

We show a mapping of the complete projective space into the Euclidean space of the same dimension, exploiting the stereographic projection which guarantees that straight lines of the projective space map to circles in the Euclidean space.

36.1 The Task

We use the stereographic projection to map the complete projective space into the Euclidean space. We want to show, the mapping maps straight lines of the projective space to circles in the Euclidean space.

As an example take the projective plane $\mathbb{P}^2$ with the origin $\mathbf{x}_O$, its two points $\mathbf{x}_{\infty x}$ and $\mathbf{x}_{\infty y}$ at infinity in x- and y- direction, the coordinate axes $\boldsymbol{\ell}_x$ and $\boldsymbol{\ell}_y$ and the line $\boldsymbol{\ell}_\infty$ at infinity: They can be visualized as in Fig. 36.1, see Fig. 5.27 in (Förstner and Wrobel, 2016, p. 244).

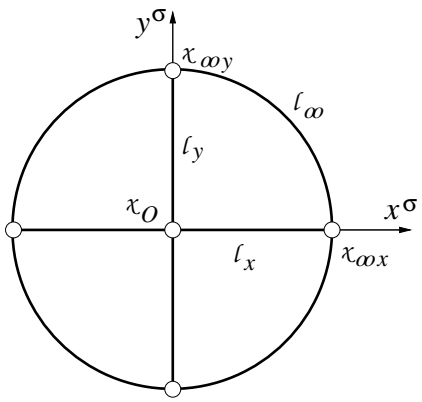


Figure 36.1: The basic elements of the projective plane $\mathbb{P}^2$

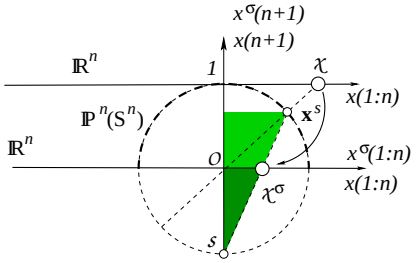


Figure 36.2: Stereographic mapping of the projective plane for visualization. Mapping the unit sphere S^2 representing $\mathbb{P}^2$ onto the unit disc in the equator plane using a stereographic projection. Each point $\mathcal{X}$ is mapped to $\mathcal{X}^\sigma$ via the spherically normalized point $\mathbf{x}^s$ on the upper half of the unit sphere, seen from the south pole s of the unit sphere. The complete projective plane P with positive points $\mathcal{X}([u, v, w]), w > 0$ is mapped into the interior of the great circle in the (x^σ, y^σ)-plane.

36.2 The mapping

Let the homogeneous coordinates of $\mathcal{X}(\mathbf{x}) \in \mathbb{P}^d$ with homogeneous coordinates

$$\begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix} := \begin{bmatrix} x_1 \\ \dots \\ x_\iota \\ \dots \\ x_d \\ x_{d+1} \end{bmatrix} = \mathbf{x} \tag{36.1}$$

We assume the points to be positive, thus $x_h = x(d+1) > 0$. Points with negative last coordinate are called negative points.

Spherical normalization leads to

$$\mathbf{x}^s = \frac{1}{|\mathbf{x}|} \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix}. \tag{36.2}$$

Then we have the stereographically mapped point at (see the similarity of the two green triangles)

$$\mathbf{x}^\sigma = \begin{bmatrix} \frac{\boldsymbol{x}_0^s}{1+x_h^s} \\ 1 \end{bmatrix} \simeq \begin{bmatrix} \boldsymbol{x}_0 \\ x_h + |\mathbf{x}| \end{bmatrix} \tag{36.3}$$

The complete projective space is mapped into the unit ball. Inversion yields

$$\mathbf{x}(\mathbf{x}^\sigma) = \begin{bmatrix} 2x_h^\sigma \boldsymbol{x}^\sigma \\ (x_h^\sigma)^2 - |\boldsymbol{x}_0^\sigma|^2 \end{bmatrix}. \tag{36.4}$$

Remark: Points outside the unit ball have their pre-image in negative points, thus with points having homogeneous coordinate $x(n+1) < 0$.

If we start from Cartesian coordinates $\boldsymbol{x}$ using $\mathbf{x} = [\boldsymbol{x}; 1]$ we therefore can write the mapping as

$$\boldsymbol{x}^\sigma(\boldsymbol{x}) = \frac{\boldsymbol{x}}{1 + \sqrt{1 + |\boldsymbol{x}|^2}}. \tag{36.5}$$

All points map into the interior of the unit ball. Inversion yields

$$\boldsymbol{x}(\boldsymbol{x}^\sigma) = \frac{2}{1 - |\boldsymbol{x}^\sigma|^2}\, \boldsymbol{x}^\sigma \tag{36.6}$$

36.3 Example

Let us assume the area of interest is the square $[-1,+1]^2$, shown in black in Figure 36.4 upper left. The major region is in the square $[-40,+40]^2$ shown in blue. There however may be objects outside this region.

Specifying a reference radius R we first apply a down scaling and afterwards an up-scaling using the matrix

$$S(R) = \begin{bmatrix} R\,I_n & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} . \tag{36.7}$$

This yields the mapping

$$\mathbf{x}^\sigma = S(R)\,\mathbf{x}^\sigma \left(S\left(\frac{1}{R}\right)\mathbf{x} \right) \quad \text{or} \quad \boldsymbol{x}^\sigma = R\,\boldsymbol{x}^\sigma\left(\frac{\boldsymbol{x}}{R}\right) . \tag{36.8}$$

This maps the complete projective space into the sphere with radius R.

Distorted projections are shown in Figure 36.4 for different reference radii.

Using an quadtree in the bounding box $[-R,+R]$ for accessing the original elements of the scene allows to handle elements at infinity easily.

The idea can be transferred to 3D: Then the complete projective 3D space, including all elements at infinity, e.g., the sky, can be addressed using an octree centred at the object of interest.

36.4 Proof

The proof uses the fact that the projection of a point $\mathcal{X}$ to the unit sphere maps is a gnomonic projection, which maps straight lines to great circles, which then are mapped to circles using the stereographic projection.

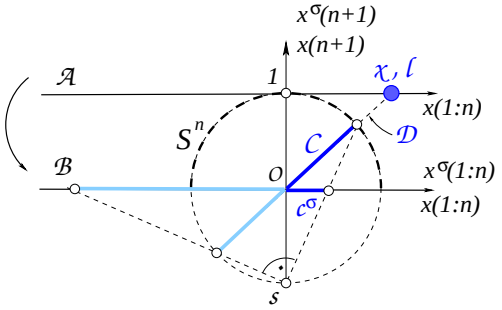


Figure 36.3: Stereographic mapping of the projective plane for visualization. The mapping maps straight lines into circles. This can be seen by decomposing the mapping into (1) a gnomonic mapping from the plane $\mathcal{A} : x(n+1) = 1$ to the unit sphere S^2, and (2) a stereographic mapping of the unit sphere S^2 onto the plane $\mathcal{B} : x(n+1) = x^\sigma(n+1) = 0$.

The proof for arbitrary n is the following:

1. The point $\mathcal{X} \in \mathbb{R}^n$ (2D point $n = 2$ in the horizontal plane $x(n+1) = 1$) is mapped to the unit sphere S^n (S^2) using a gnomonic projection.
   The gnomonic projection maps a hyperplane ℓ (straight line at the blue point $\mathcal{X}$ being perpendicular to the drawing plane) in $\mathbb{R}^n$ (R^2) to a hypersphere $\mathcal{C}$ (blue great circle) on the unit sphere S^n (S^2) in R^{n+1}: It is the intersection of the hyperplane (plane) $\mathcal{D} = \mathcal{O} \wedge \ell$ spanned by the origin $\mathcal{O}$ and ℓ. This hyperplane (plane) $\mathcal{D}$ intersects the unit sphere S^n in a great hypersphere $\mathcal{C}$ on S^n. Only the blue semicircle above the origin is valid. The light blue circle below results from negative points on the line ℓ.

2. A stereographic projection of S^n to R^n is sphere-preserving. Therefore, also the hypersphere $\mathcal{C}$ on S^n maps to a sphere c^σ (the blue circle in Figure 36.3) in $\mathbb{R}^n$ ($\mathbb{R}^2$ here sitting in the horizontal plane through O). In this special case only the image right is valid. The light grey part of the circle results from negative points on the line ℓ.

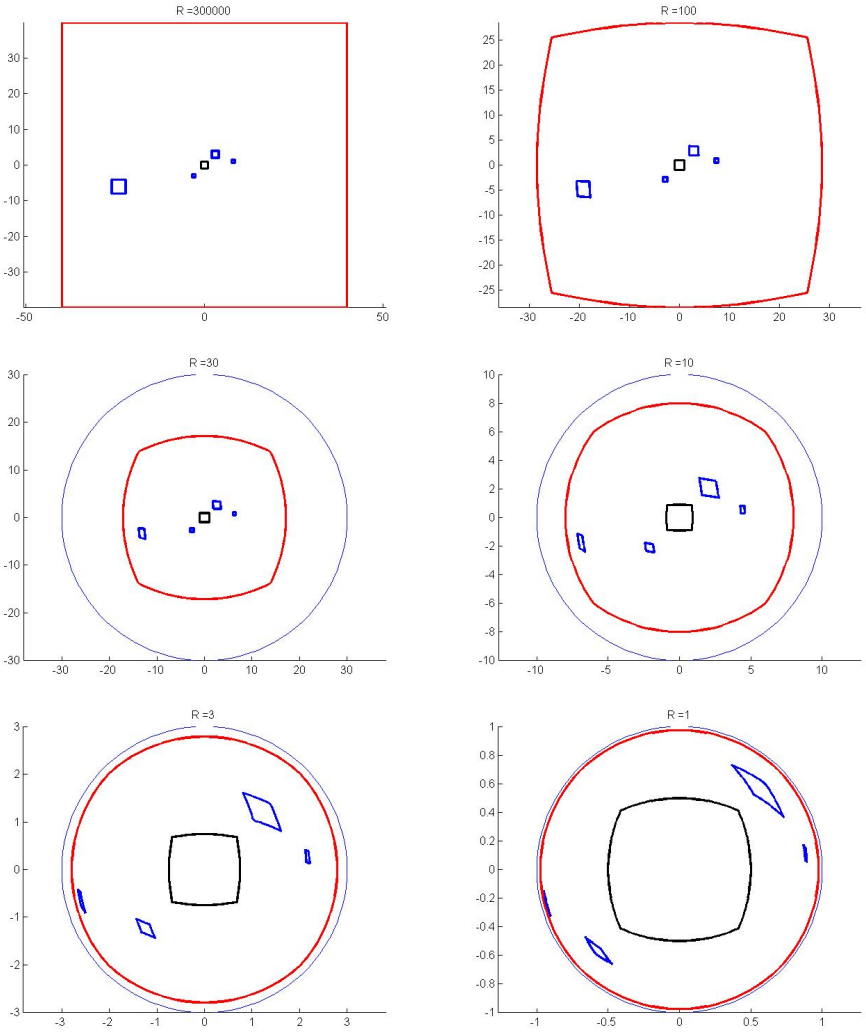


Figure 36.4: Boxes stereographically distorted with different reference radii. The upper subfigure with $R = 100$ shows only small distortions. The lower subfigures with $R = 30, 10, 3, 1$ show more and more distortions together with the reference circle with radius R. In all cases the borders of the boxes are circle segments

Part VI

Technical Notes on Bundle Adjustment and Surface Reconstruction

37 Rotation from the Essential Matrix for Zero-Basis

We show, that the determination of the rotation and translation from an estimated the essential matrix for an image pair yields the correct rotation matrix in case the basis is 0.

37.1 Preface

This note (2011) shows, that the determination of the rotation and translation, when estimating the essential matrix for an image pair yields the correct rotation matrix in case the basis is 0.

37.2 The Problem

Given N corresponding points $(\mathbf{x}'_i, \mathbf{x}''_i), i = 1, ..., I$ in a situation, where the basis has zero length. The task is to determine the essential matrix

$$\mathsf{E} = S(\mathbf{b})R^{\mathsf{T}} \tag{37.1}$$

based on the epipolar constraints

$$0 = {\mathbf{x}_i}'^{\mathsf{T}}\mathsf{E}\mathbf{x}''_i = ({\mathbf{x}_i}'^{\mathsf{T}} \otimes {\mathbf{x}_i}''^{\mathsf{T}})\mathbf{e} \quad \text{with} \quad \mathbf{e} = \mathrm{vec}\mathsf{E}\,. \tag{37.2}$$

Empirically, in case the basis is zero, the null space of the matrix

$$A = [{\mathbf{x}_i}''^{\mathsf{T}} \otimes {\mathbf{x}_i}'^{\mathsf{T}}] \tag{37.3}$$

is six. However, the rotation derived from any vector in the null space of A is correct. This is plausible. A proof appears to be missing. This note provides a proof.

37.3 Proof

The idea of the proof is the following: In case the basis is zero we have

$$\mathbf{x}''_i = R\mathbf{x}'_i\,. \tag{37.4}$$

Therefore the matrix A from (31.11) specializes to

$$A = [{\mathbf{x}_i}'^{\mathsf{T}}R^{\mathsf{T}} \otimes {\mathbf{x}_i}'^{\mathsf{T}}] = [({\mathbf{x}_i}'^{\mathsf{T}} \otimes {\mathbf{x}_i}'^{\mathsf{T}})(R^{\mathsf{T}} \otimes I_3)] = A_r(R^{\mathsf{T}} \otimes I_3)\,. \tag{37.5}$$

On the other hand the essential matrix with $\mathbf{b} \neq \mathbf{0}$

$$\mathsf{E} = S(\mathbf{b})R^\mathsf{T} = [\boldsymbol{s}_1 \mid \boldsymbol{s}_2 \mid \boldsymbol{s}_3]R^\mathsf{T} \tag{37.6}$$

has vector

$$\mathbf{e} = (R \otimes I_3)\mathrm{vec}(\mathsf{S}(\mathbf{b}))\,. \tag{37.7}$$

Therefore we have, independent on the choice of $\mathbf{b}$

$$\begin{aligned}
A\mathbf{e} &= \left[(\mathbf{x}_i'^\mathsf{T} \otimes \mathbf{x}_i'^\mathsf{T})(R^\mathsf{T} \otimes I_3)\right](R \otimes I_3)\mathrm{vec}(\mathsf{S}(\mathbf{b})) && (37.8)\\
&= [\mathbf{x}_i'^\mathsf{T} \otimes \mathbf{x}_i'^\mathsf{T}]\mathrm{vec}(\mathsf{S}(\mathbf{b})) && (37.9)\\
&= [\mathbf{x}_i'^\mathsf{T}\mathsf{S}(\mathbf{b})\mathbf{x}_i'] && (37.10)\\
&= [|\mathbf{x}_i' \mid \mathbf{b} \mid \mathbf{x}_i'|] && (37.11)\\
&= [0]\,. && (37.12)
\end{aligned}$$

Thus, whatever value $\mathbf{b}$ has, the epipolar constraint is fulfilled, if we have a pure rotation. This is important, since algorithms for determining E and derive $(\mathbf{b}, R)$ enforce $\mathbf{b}$ to have length 1.

The rank of A in (37.5) is 6 for a general configuration of points $\mathbf{x}_i$, since the rank of A_r is 6

$$\begin{aligned}
\mathrm{rk}A = \mathrm{rk}A_r &= \mathrm{rk}[u_i^2 \mid u_iv_i \mid u_iw_i \mid v_iu_i \mid v_i^2 \mid v_iw_i \mid w_iu_i \mid w_iv_i \mid w_i^2] && (37.13)\\
&= \mathrm{rk}[\underbrace{u_i^2 \mid v_i^2 \mid w_i^2}_{\boldsymbol{\alpha}^\mathsf{T}} \mid \underbrace{u_iv_i \mid u_iw_i \mid v_iw_i}_{\boldsymbol{\beta}^\mathsf{T}} \mid \underbrace{u_iv_i \mid u_iw_i \mid v_iw_i}_{\boldsymbol{\beta}^\mathsf{T}}] && (37.14)\\
&= \mathrm{rk}[\underbrace{\boldsymbol{\alpha}_i^\mathsf{T} \mid \boldsymbol{\beta}_i^\mathsf{T}}_{\mathrm{rk}=6} \mid \boldsymbol{\beta}_i^\mathsf{T}] && (37.15)\\
&= 6 && (37.16)
\end{aligned}$$

Any vector $\mathbf{e} = \mathrm{vec}(\mathsf{E}) \in \mathsf{null}(A)$ in is of the form

$$\mathbf{e} = (R \otimes I_3)\mathrm{vec}(\mathsf{S}(\mathbf{b}))) \tag{37.17}$$

with the correct rotation matrix and an arbitrary $\mathbf{b}$, explaining the loss in rank from 8 to 6 for noise-less data. Thus, the partitioning of E, taking into account the visibility of the points, leads to the correct rotation matrix.

For noisy data we will obtain an approximation for the correct rotation.

37.4 Optimal solution for the rotation for zero basis

The above-mentioned solution is not optimal in any sense, as the normalization of the image point vectors may be arbitrary.

The optimal solution, see Förstner and Wrobel (2016, Sect. 13.3.5.1), for the relative orientation with non-zero basis uses the representation

$$\mathsf{E} = \mathsf{S}(\mathbf{b})R^\mathsf{T} \tag{37.18}$$

and the constraint

$$g_i(\mathbf{x}_i', \mathbf{x}_i'', \mathbf{b}, R) = \mathbf{x}_i'^\mathsf{T}\mathsf{S}(\mathbf{b})R^\mathsf{T}\mathbf{x}_i'' \tag{37.19}$$

valid for the true values. The Jacobians are

$$\boldsymbol{a}_i^\mathsf{T} = [({}^1\mathbf{x}_i''^\mathsf{T} \times \mathbf{x}_i'^\mathsf{T})J_r(\mathbf{b}') \mid \mathbf{l}_i''^\mathsf{T} \times \mathbf{x}_i''^\mathsf{T}] \doteq [\boldsymbol{a}_{ip}^\mathsf{T} \mid \boldsymbol{a}_{ip}^\mathsf{T}] \tag{37.20}$$

and

$$\mathbf{b}_i^\mathsf{T} = [\mathbf{l}_i'^\mathsf{T}J_r(\mathbf{x}_i')'^\mathsf{T} \mid \mathbf{l}_i''^\mathsf{T}J_r(\mathbf{x}_i'')] \tag{37.21}$$

with

$$\mathbf{l}'_i = \mathsf{E}\mathbf{x}''_i \qquad \mathbf{l}''_i = \mathsf{E}^\mathsf{T}\mathbf{x}'_i \qquad {}^1\mathbf{x}''_i = R^\mathsf{T}\mathbf{x}''_i \tag{37.22}$$

all expressions evaluated at the approximated values. The covariance matrix of the observations is

$$\mathbb{D}\left(\begin{bmatrix} \mathbf{x}'_i \\ \mathbf{x}''_i \end{bmatrix}\right) = \Sigma_{y_i y_i} \tag{37.23}$$

The normal equation matrix reads

$$N = \begin{bmatrix} N_{bb} & N_{bp} \\ N_{pb} & N_{pp} \end{bmatrix} = \begin{bmatrix} \sum_i w_i \boldsymbol{a}_{ib}\boldsymbol{a}_{ib}^\mathsf{T} & \sum_i w_i \boldsymbol{a}_{ib}\boldsymbol{a}_{ip}^\mathsf{T} \\ \sum_i w_i \boldsymbol{a}_{ip}\boldsymbol{a}_{ib}^\mathsf{T} & \sum_i w_i \boldsymbol{a}_{ip}\boldsymbol{a}_{ip}^\mathsf{T} \end{bmatrix} \tag{37.24}$$

with

$$w_i = \mathbf{b}_i^\mathsf{T}\Sigma_{y_i y_i}\boldsymbol{b}_i\,. \tag{37.25}$$

The right hand sides are

$$c_{g_i}(\mathbf{x}'_i, \mathbf{x}''_i, \mathbf{b}^a, R^a) = -\mathbf{x}_i'^{a\mathsf{T}}\mathsf{S}(\mathbf{b}^a)R^{a\mathsf{T}}\mathbf{x}_i''^a + \boldsymbol{b}_i^\mathsf{T}\boldsymbol{v}_i \tag{37.26}$$

In case the basis is zero, the vectors $\boldsymbol{a}_{ib} = {}^1\mathbf{x}''_i \times \mathbf{x}'_i$ are zero. Therefore, we cannot determine the direction of the basis. However, the correct rotation parameters for the rotation matrix $R^{(\nu)}$ in the ν-th iteration

$$R^{(\nu+1)} = R(\Delta\boldsymbol{p})R^{(\nu)} \tag{37.27}$$

result from

$$N_{pp}\Delta\boldsymbol{p} = \boldsymbol{h}_p \tag{37.28}$$

with

$$N_{pp} = \sum_i w_i(\mathbf{l}''_i \times \mathbf{x}''_i)(\mathbf{l}_i''^\mathsf{T} \times \mathbf{x}_i''^\mathsf{T}) \qquad \boldsymbol{h}_p = \sum_i w_i(\mathbf{l}''_i \times \mathbf{x}''_i)c_{g_i} \tag{37.29}$$

and theoretical covariance matrix (Cramer-Rao bound)

$$\Sigma_{rr} = N_{pp}^{-1}\,. \tag{37.30}$$

38 Rule of Thumb for Precision of Points from Multiview Triangulation

For planning bundle adjustment configurations, the expected accuracy of triangulated points is an essential ingredient. We derive rules of thumb for the accuracy of multi-view triangulating by providing simple expressions for the depth and lateral accuracy of 3D points, for images arranged in a line, in a planar region and in a spherical region, covering the case of omnidirectional cameras.

38.1 Preface

The note (2013) provides explicit expressions (rules of thumb) for the depth accuracy obtained from multi-view triangulation for three cases: (1) the projection centers lie in a line, (2) the projection centers lie in square, and (3) the projection centers are equally spaced on a spherical cap. The note is the basis for Förstner and Wrobel (2016, Sect. 15.7.1).

38.2 Problem

Given T images of a 3D point determine the precision of its position.

The standard deviation depends on

1. on whether the projection centers are in a row, in a rectangular grid, or on a spherical cap
2. the coordinate precision $\sigma_{x'}$ or the directional precision σ_α,
3. the principal distance c,
4. the baseline B or the diameter D of the set of projection centers, on the spherical cap δ measured in radiants, and
5. the common height Z above the unknown point or the radius Z of the spherical cap.

If the T projection centers are in a row we have for large T

$$\boxed{\sigma_{\widehat{W}}^{(1D)} = \frac{\sqrt{12}}{T^{3/2}} \frac{Z^2}{B} \frac{\sigma_{x'}}{c} = \sqrt{\frac{12}{T}} \frac{Z^2}{D} \frac{\sigma_{x'}}{c}.} \tag{38.1}$$

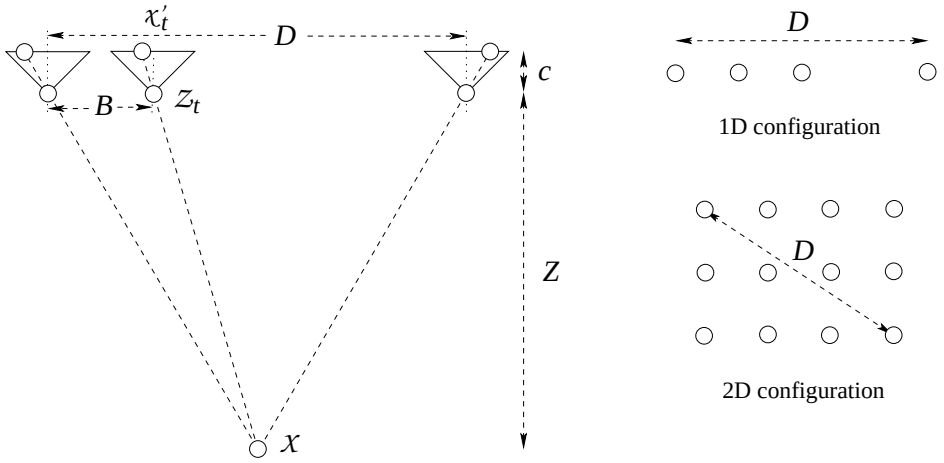


Figure 38.1: Ideal configuration for triangulation. Alternatively, the projection centers are on a sphere with radius Z regularly spaced in a spherical cap with diameter δ.

If the T projection centers are in a rectangular grid we have for large T

$$\boxed{\sigma_{\widehat{W}}^{(2D)} = \frac{\sqrt{6}}{T}\frac{Z^2}{B}\frac{\sigma_{x'}}{c} = \sqrt{\frac{12}{T}}\frac{Z^2}{D}\frac{\sigma_{x'}}{c}}, \tag{38.2}$$

If the T projection centers are evenly distributed on a spherical cap with diameter δ we have

$$\boxed{\sigma_{\widehat{W}}^{(\text{cap})} = \frac{\sqrt{3}}{\sqrt{T}}\frac{Z}{2-\cos\frac{\delta}{2}-\cos^2\frac{\delta}{2}}\sigma_\alpha\,.} \tag{38.3}$$

38.3 Formal statement

Without loss of generality the scene coordinate system sits close to the unknown scene point $\mathcal{X}([U,V,W])$. It is observed in T cameras, which for simplicity are assumed to be identical and are nadir views with $\mathsf{R} = \mathsf{I}_3$. Their common principal distance is c. Their projection centers $\mathcal{Z}_t$ are at $\boldsymbol{Z}_t, t = 1,...,T$. The projection matrices therefore are

$$\mathsf{P}_t = \text{Diag}([c,c,1])[\mathsf{I}_3|-\boldsymbol{Z}_t]\,. \tag{38.4}$$

We observe the T image points

$$\boldsymbol{x}'_t = c\begin{bmatrix} x_t \\ y_t \end{bmatrix} = c\frac{1}{U-Z_t}\begin{bmatrix} V-X_t \\ W-Y_t \end{bmatrix}. \tag{38.5}$$

The task is to estimate the unknown parameters $\boldsymbol{X}$.

38.4 Linearization

Using $\boldsymbol{X}^a = \boldsymbol{0}$, the linearized model reads as

$$\Delta\boldsymbol{x}'_t = c\begin{bmatrix} -\frac{1}{Z_t} & 0 & \frac{X_t}{Z_t^2} \\ 0 & -\frac{1}{Z_t} & \frac{Y_t}{Z_t^2} \end{bmatrix}\begin{bmatrix} \Delta U \\ \Delta V \\ \Delta W \end{bmatrix}. \tag{38.6}$$

With weights w_t for each point we obtain the normal equation matrix

$$\mathsf{N} = c^2\begin{bmatrix} \sum_t \frac{w_t}{Z_t^2} & 0 & -\sum_t \frac{w_t X_t}{Z_t^3} \\ 0 & \sum_t \frac{w_t}{Z_t^2} & -\sum_t \frac{w_t Y_t}{Z_t^3} \\ -\sum_t \frac{w_t X_t}{Z_t^3} & -\sum_t \frac{w_t X_t}{Z_t^3} & \sum_t \frac{w_t(X_t^2+Y_t^2)}{Z_t^4} \end{bmatrix} \tag{38.7}$$

If we assume the projection centers have the same Z-coordinate we obtain

$$\mathsf{N} = \frac{c^2}{Z^4}\begin{bmatrix} \sum_t w_t Z^2 & 0 & -\sum_t w_t X_t Z \\ 0 & \sum_t w_t Z_t^2 & -\sum_t w_t Y_t Z \\ -\sum_t w_t X_t Z & -\sum_t w_t X_t Z & \sum_t w_t (X_t^2 + Y_t^2) \end{bmatrix} \tag{38.8}$$

If we now assume the X- and Y-coordinates are centred with

$$\bar{X} = \frac{\sum_t w_t X_t}{\sum_t w_t} \quad \bar{Y} = \frac{\sum_t w_t Y_t}{\sum_t w_t} \tag{38.9}$$

and the weights are constant

$$w = \frac{1}{\sigma_{x'}^2} \tag{38.10}$$

the normal equation matrix is diagonal

$$\mathsf{N} = \frac{c^2}{Z^4 \sigma_{x'^2}}\begin{bmatrix} TZ^2 & 0 & 0 \\ & TZ^2 & 0 \\ 0 & 0 & \sum_t (X_t^2 + Y_t^2) \end{bmatrix}. \tag{38.11}$$

If we use the average distance of the projection center from its centroid

$$S = \sqrt{\frac{\sum_t (X_t^2 + Y_t^2)}{T}} \tag{38.12}$$

of the projection centers it reads as

$$\mathsf{N} = \frac{c^2}{Z^4 \sigma_{x'^2}}\begin{bmatrix} TZ^2 & 0 & 0 \\ & TZ^2 & 0 \\ 0 & 0 & S^2 T \end{bmatrix}. \tag{38.13}$$

Thus the variances of the 3D point are

$$\sigma_{\widehat{U}} = \sigma_{\widehat{V}} = \frac{Z}{\sqrt{T}} \frac{\sigma_{x'}}{c} \quad \text{and} \quad \sigma_{\widehat{W}} = \frac{Z^2}{S\sqrt{T}} \frac{\sigma_{x'}}{c} = \frac{Z}{S} \sigma_{\widehat{U}}\,. \tag{38.14}$$

38.5 Special configurations

38.5.1 Projection centers are on a straight line

If the T projection centers are on a straight line with basis B in X-direction, their X_t-coordinates are

$$X_t = \left(t - \frac{T+1}{2}\right) B \qquad t = 1, ..., T \quad \text{with} \quad -X_1 = X_T = \frac{T-1}{2} B\,. \tag{38.15}$$

Then we have

$$S^2 = \frac{1}{12}(T^2 - 1) B^2\,. \tag{38.16}$$

Thus we obtain the standard deviation

$$\sigma_{\widehat{W}} = \frac{\sqrt{12}}{\sqrt{T(T^2-1)}} \frac{Z^2}{B} \frac{\sigma_{x'}}{c}\,. \tag{38.17}$$

For large T we can use the approximation

$$\boxed{\sigma_{\widehat{W}}^{(1D)} = \frac{\sqrt{12}}{T^{3/2}} \frac{Z^2}{B} \frac{\sigma_{x'}}{c}\,.} \tag{38.18}$$

If we use as reference the diameter D of the projection centers

$$D = (T-1)B \tag{38.19}$$

the average distance is

$$S^2 = \frac{1}{12}\frac{T+1}{T-1}D^2\,. \tag{38.20}$$

and the standard deviation is

$$\sigma_{\widehat{W}} = \sqrt{\frac{12(T-1)}{T(T+1)}}\frac{Z^2}{D}\frac{\sigma_{x'}}{c}\,. \tag{38.21}$$

which for large T simplifies to

$$\boxed{\sigma_{\widehat{W}}^{(1D)} = \sqrt{\frac{12}{T}}\frac{Z^2}{D}\frac{\sigma_{x'}}{c}\,.} \tag{38.22}$$

38.5.2 Projection centers are on a regular grid

If the $T = MN$ projection centers are on a regular grid with basis B_X in X- and B_Y in Y direction, their coordinates are

$$X_m = \left(m - \frac{M+1}{2}\right)B_X \qquad m = 1, ..., M \quad \text{and} \quad Y_n = \left(n - \frac{N+1}{2}\right)B_Y \qquad n = 1, ..., N\,. \tag{38.23}$$

Then we have

$$S^2 = S_X^2 + S_Y^2 = \frac{1}{12}((M^2-1)B_X^2 + (N^2-1)B_Y^2)\,. \tag{38.24}$$

We now assume the grid is quadratic with $B_X = B_Y$ and $T = N^2$. Then we obtain

$$S^2 = S_X^2 + S_Y^2 = \frac{1}{6}(N^2-1)B^2 = \frac{1}{6}(T-1)B^2\,. \tag{38.25}$$

Then the standard deviation is

$$\sigma_{\widehat{W}} = \frac{\sqrt{6}}{\sqrt{T(T-1)}}\frac{Z^2}{B}\frac{\sigma_{x'}}{c}\,. \tag{38.26}$$

For large T we can use the approximation

$$\boxed{\sigma_{\widehat{W}}^{(2D)} = \frac{\sqrt{6}}{T}\frac{Z^2}{B}\frac{\sigma_{x'}}{c}}\,. \tag{38.27}$$

Using the diameter

$$D = \sqrt{2}(N-1)B \tag{38.28}$$

we have the average distance squared

$$S^2 = \frac{1}{12}\frac{T-1}{\sqrt{T}-1}D^2 \tag{38.29}$$

which yields the standard deviation

$$\sigma_{\widehat{W}}^{(2D)} = \sqrt{\frac{12(\sqrt{T}-1)}{T-1}}\frac{Z^2}{D}\frac{\sigma_{x'}}{c} \tag{38.30}$$

which for large T simplifies to

$$\boxed{\sigma_{\widehat{W}}^{(2D)} = \sqrt{\frac{12}{T}}\frac{Z^2}{D}\frac{\sigma_{x'}}{c}}\,, \tag{38.31}$$

which is identical to the standard deviation if the projection centers are on a straight line.

38.5.3 Projection centers on a spherical cap

If the T projection centers are evenly distributed on a spherical cap with radius Z and angular diameter δ we use a slightly different model. We assume the uncertainty of the rays to be uniform in all directions with standard deviation σ_α, which corresponds to $\sigma_{x'}/c$ if the observed point is close to the principal point. Then the uncertainty of the ray at the observed image point is $\sigma_q = Z\sigma_\alpha$. The direction of the ray is

$$\boldsymbol{d} = \begin{bmatrix} \cos\lambda\sin\phi \\ \sin\lambda\sin\phi \\ \cos\phi \end{bmatrix} . \tag{38.32}$$

The normal equation matrix is (see PCV-A Sect. 9.5.3.2)

$$\mathsf{N} = \sum_t w_t (\mathsf{I}_3 - \boldsymbol{d}_t \boldsymbol{d}_t^\mathsf{T}) . \tag{38.33}$$

We again assume $w_t = 1/\sigma_q^2$.

We now replace the sum by an integral

$$\mathsf{N} = w_t \sum_t (\mathsf{I}_3 - \boldsymbol{d}_t \boldsymbol{d}_t^\mathsf{T}) \approx T \frac{1}{\sigma_q^2} \frac{\int_{\lambda,\phi\in C} (\mathsf{I}_3 - \boldsymbol{d}_t \boldsymbol{d}_t^\mathsf{T}) \cos\phi \, \mathrm{d}\lambda\mathrm{d}\phi}{\int_{\lambda,\phi\in C} \cos\phi \mathrm{d}\lambda\mathrm{d}\phi} . \tag{38.34}$$

For symmetry reason the normal equation matrix is diagonal:

$$N_{11} = N_{22} = \frac{1}{6\sigma_q^2}\left(4 + \cos^2\frac{\delta}{2} + \cos\frac{\delta}{2}\right) T \quad \text{and} \quad N_{33} = \frac{1}{3\sigma_q^2}\left(2 - \cos^2\frac{\delta}{2} - \cos\frac{\delta}{2}\right) T , \tag{38.35}$$

the second expression proving (38.3).

Observe for $d = 2\pi$ due to $\cos\frac{\delta}{2} = -1$ we obtain the fully isotropic configuration

$$N_{11} = N_{22} = N_{33} = \frac{2}{3\sigma_q^2} . \tag{38.36}$$

Thus the standard deviation for the ZW-coordinate is

$$\sigma_{\widehat{W}}^{(\mathrm{cap})} = \frac{\sqrt{3}}{\sqrt{T}} \frac{Z}{2 - \cos\frac{\delta}{2} - \cos^2\frac{\delta}{2}} \sigma_\alpha . \tag{38.37}$$

For small δ we obtain the approximation

$$\sigma_{\widehat{W}}^{(\mathrm{cap})} = \frac{\sqrt{8}}{\sqrt{T}} \frac{Z}{\delta} \sigma_\alpha . \tag{38.38}$$

Taking into account that then $\delta = D/Z$ and $\sigma_\alpha = \sigma_{x'}/c$ we obtain

$$\sigma_{\widehat{W}}^{(\mathrm{cap})} = \frac{\sqrt{8}}{\sqrt{T}} \frac{Z^2}{D} \frac{\sigma_{x'}}{c} . \tag{38.39}$$

The difference of the constants ($\sqrt{12}$ versus $\sqrt{8}$) result from the different roundness of the two figures (square versus circle).

39 Multi-View Triangulation with Directions

We provide simple solution to the optimal triangulation of a scene point from multiple views assuming isotropic uncertainty of the directions. As a special case we provide a simple expression for the distance of the triangulated point in case of homogeneous directional uncertainty and small basis, expressed as a function of the effective base line, the viewing angle and the resolution of an omnidirectional camera and the matching accuracy in pixels.

39.1 Preface

This note from 2007, and extended 2023, provides a simple solution to the optimal triangulation of a scene point from multiple views. It also provides a simple expression for the distance of the triangulated point in case of homogeneous directional uncertainty and small basis, expressed as a function of the effective base line, the viewing angle and the resolution of an omnidirectional camera and the matching accuracy in pixels.

39.2 The Problem

Given are K projection matrices $\mathsf{P}_k, k = 1, ..., K$ and corresponding image points $\boldsymbol{x}_k, k = 1, ..., K$. Triangulate a good 3D-point.k The idea is the following: The projection matrices together with the image point determine N projection rays, see Fig. 39.1. The optimal point $\boldsymbol{X}$ is the one closest to all these rays, where the notion distance needs to be specified and leads to different solutions.

We extend the approximate solution in three ways:

1. We handle the case where the distances are weighted individually.
2. We handle the case of isotropic and homogeneous uncertainty of the directions.
3. We handle the case of homogeneous mutual correlations between the directions.

In all cases we provide a rigorous solution.

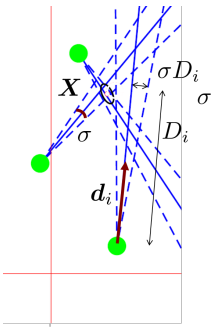


Figure 39.1: Optimal multi-view triangulation for directions. The problem is nonlinear in general, since the effect of directional uncertainties onto the 3D point depends on the unknown distances of the point to the given projection centers

39.3 The approximate Solution

The first solution just minimizes the sum of the squares of the distances of the rays to the 3D point.

The projection centers are

$$\boldsymbol{Z}_k = -\mathsf{H}_{n\infty}^{-1}\mathbf{h}_k \tag{39.1}$$

with

$$\mathsf{P}_k = [\mathsf{H}_{n\infty}|\mathbf{h}_k] \tag{39.2}$$

The projection lines have normalized direction

$$\mathbf{d}_k = \mathsf{N}\left(\mathsf{H}_{n\infty}^{-1}\mathbf{x}_k\right) \tag{39.3}$$

The 3D projection lines have Plücker coordinates

$$\mathbf{L}_k = \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix}_k = \begin{bmatrix} \mathbf{d}_k \\ \boldsymbol{Z}_k \times \mathbf{d}_k \end{bmatrix} \tag{39.4}$$

The squared distances of the unknown point $\boldsymbol{X}$ to the lines are

$$\begin{aligned} d_{XL_k}^2 &= |\boldsymbol{L}_{0i} + \mathcal{S}(\boldsymbol{L}_{hi})\boldsymbol{X}|^2 && (39.5)\\ &= (\boldsymbol{Z}_k \times \mathbf{d}_k + \mathcal{S}(\mathbf{d}_k)\boldsymbol{X})^\mathsf{T}(\boldsymbol{Z}_k \times \mathbf{d}_k + \mathcal{S}(\mathbf{d}_k)\boldsymbol{X}) && (39.6)\\ &= |\boldsymbol{Z}_k \times \mathbf{d}_k|^2 + 2(\boldsymbol{Z}_k \times \mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{X} + \boldsymbol{X}^\mathsf{T}\mathcal{S}(\mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{X} && (39.7) \end{aligned}$$

The sum of the squared distances therefore is

$$\begin{aligned} \Omega &= \sum_k d_{XL_k}^2 && (39.8)\\ &= \sum_k |\boldsymbol{Z}_k \times \mathbf{d}_k|^2 + 2\sum_k (\boldsymbol{Z}_k \times \mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{X} + \boldsymbol{X}^\mathsf{T}\sum_k \mathcal{S}(\mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{X} && (39.9) \end{aligned}$$

The necessary condition for the minimum is

$$\frac{1}{2}\frac{\partial \Omega}{\partial \boldsymbol{X}} = \sum_k \mathcal{S}(\mathbf{d}_k)^\mathsf{T}(\boldsymbol{Z}_k \times \mathbf{d}_k) + \sum_k \mathcal{S}(\mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{X} = \mathbf{0} \tag{39.10}$$

Thus, the optimal point is given by

$$\widehat{\boldsymbol{X}} = \left(\sum_k \mathcal{S}(\mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\right)^{-1} \sum_k \mathcal{S}(\mathbf{d}_k)^\mathsf{T}\mathcal{S}(\mathbf{d}_k)\boldsymbol{Z}_k \tag{39.11}$$

or

$$\boxed{\widehat{\boldsymbol{X}} = \left(\sum_k W_k\right)^{-1} \sum_k W_k \boldsymbol{Z}_k} \tag{39.12}$$

with

$$W_k = I_3 - \mathbf{d}_k \mathbf{d}_k^\mathsf{T} \tag{39.13}$$

in case $\mathbf{d}_k$ is normalized. Obviously, this is a weighted mean of the projection centers $\boldsymbol{Z}_k$ where the weight matrix is 0 in the direction of $\mathbf{d}_k$ and 1 otherwise. Thus, W_k is representing a cylindrical covariance matrix, with infinite uncertainty in the direction of the projection lines and standard deviation 1 perpendicular to the viewing direction.

The estimated variance of the distances of the fitted points to the projection lines can be obtained from

$$\boxed{\hat{\sigma}_d^2 = \frac{\Omega}{2I-3} \quad \text{with} \quad \Omega = \sum_k d^2(\mathcal{X}, \mathcal{L}_k) = \sum_k \left(\frac{\mathbf{d}_k^\mathsf{T}(\widehat{\boldsymbol{X}} - \boldsymbol{Z}_k)}{|\widehat{\boldsymbol{X}} - \boldsymbol{Z}_k|}\right)^2 .} \tag{39.14}$$

The theoretical covariance matrix of the estimated points is

$$\boxed{\Sigma_{\widehat{X}\widehat{X}} = \sigma_d^2 \left(\textstyle\sum_k W_k\right)^{-1}} \tag{39.15}$$

with some prior assumption about the standard deviation σ_d of the distances.

39.4 The Solution with Different Uncertainties of the Distances

Instead of (39.8) we optimize

$$\Omega = \sum_k \frac{d_{XL_k}^2}{\sigma_{d_k}^2} , \tag{39.16}$$

where the standard deviations of the distances are σ_{d_k}. We obtain the same solution (39.12) however instead of the weight-matrices in (39.13) we use

$$W_k = \frac{1}{\sigma_{d_k}^2} \left(I_3 - \mathbf{d}_k \mathbf{d}_k^\mathsf{T}\right) , \tag{39.17}$$

see PCV Eq. (10.174).

If the solution (39.12) is written with the normal equation matrix and the right-hand sides

$$N = \sum_k W_k \quad \text{and} \quad \boldsymbol{n} = \sum_k W_k \boldsymbol{Z}_k \tag{39.18}$$

(using the weights $eq : W - WLS$, assuming $\sigma_0 = 1$) we have the theoretical covariance matrix

$$\Sigma_{\widehat{X}\widehat{X}} = \sigma_0 N^{-1} . \tag{39.19}$$

Similarly, we obtain an estimate for the variance factor

$$\hat{\sigma}_0^2 = \frac{\Omega}{2I-3} \quad \text{with} \quad \Omega = \sum_k \left(\frac{\mathbf{d}_k^\mathsf{T}(\widehat{\boldsymbol{X}} - \boldsymbol{Z}_k)}{|\widehat{\boldsymbol{X}} - \boldsymbol{Z}_k|}\right)^2 . \tag{39.20}$$

39.5 The Solution for Directional Observations with Different Uncertainty

In case directions δ_k are observed, the uncertainty of the distances d_k of the unknown point to the given rays depend on the distances s_k of the point $\boldsymbol{x}$ to the projection centers $\boldsymbol{Z}_k$:

$$\sigma_{d_k} = s_k \sigma_{\delta_k} \quad \text{with} \quad s_k = |\boldsymbol{X} - \boldsymbol{Z}_k| \tag{39.21}$$

We cannot optimize (39.16) since the distances s_k depend on the unknown point.

However, see Fig. 39.2, we can iteratively update $\mathcal{X}$ by using (39.21) after an initialization with $s_k = 1$ in the first iteration. For not too large directional errors, say below 0.01 [rad] or 1 °, only a second iteration is necessary. This procedure can replace Algorithm 21

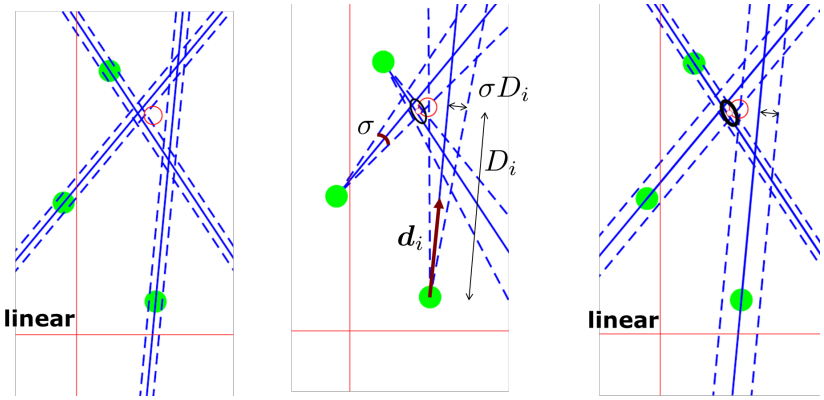


Figure 39.2: Optimal triangulation with isotropic directional uncertainties.

in PCV, in case it is clear that the 3D point is at finity and the rays do not diverge, or if some sufficiently good approximate value for $\boldsymbol{X}$ is known.

39.6 Assuming Correlations between the Directions due to Least Squares Matching

39.6.1 The 2D Model

We assume the position of the keypoint in one image is determined by some keypoint detector and the coordinate differences, i.e., parallaxes, to the other images are determined by least squares matching, like the Kanade-Lucas-Tracker. The reason simply is: the coordinates $\boldsymbol{x}_1$ of the detection usually is less accurate, say with standard deviation σ_x whereas the determination of the parallaxes $\boldsymbol{p}_k = \boldsymbol{x}_k - \boldsymbol{x}_0, k = 2, ..., K$ is highly accurate, say σ_{p_0}. Assuming a homogeneous configuration and enforcing the mean coordinate, derived from the parallaxes is $\mathbf{0}$ the uncertainty of the final coordinates $\boldsymbol{x}_k$ can be derived from

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \underline{\boldsymbol{x}}_1 \\ \dots \\ \underline{\boldsymbol{x}}_k \\ \dots \\ \underline{\boldsymbol{x}}_K \end{bmatrix} = \begin{bmatrix} \mathbb{E}(\underline{\boldsymbol{x}}_1) \\ \dots \\ \mathbb{E}(\underline{\boldsymbol{x}}_k) \\ \dots \\ \mathbb{E}(\underline{\boldsymbol{x}}_K) \end{bmatrix} + \begin{bmatrix} \underline{\Delta\boldsymbol{x}_0} \\ \dots \\ \underline{\Delta\boldsymbol{x}_0} \\ \dots \\ \underline{\Delta\boldsymbol{x}_0} \end{bmatrix} + \begin{bmatrix} \underline{\Delta\boldsymbol{p}_1} \\ \dots \\ \underline{\Delta\boldsymbol{p}_k} \\ \dots \\ \underline{\Delta\boldsymbol{p}_K} \end{bmatrix} \tag{39.22}$$

with the covariance matrices for the detection $\underline{\Delta\boldsymbol{x}_0}$ and the parallaxes $\underline{\Delta\boldsymbol{p}} = [\underline{\Delta\boldsymbol{p}_k}]$:

$$\mathbb{D}(\underline{\Delta\boldsymbol{x}_0}) = \Sigma_{x_0 x_0} \quad \text{and} \quad \mathbb{D}(\underline{\Delta\boldsymbol{p}}) = (I_K - J_K/K) \otimes \Sigma_{pp} \quad \text{with} \quad J = \mathbf{1}_K \mathbf{1}_K^\mathsf{T} \tag{39.23}$$

see Förstner (1998). In the isotropic case we have

$$\Sigma_{x_0 x_0} = \sigma_{x_0}^2 I_2 \quad \text{and} \quad \Sigma_{pp} = \sigma_p^2 I_2 \,. \tag{39.24}$$

This yields the following covariance matrix for the K points

$$\Sigma_{xx} = \left(\sigma_{x_0}^2 \, J_K + \sigma_p^2 \, (I_K - J_K/K)\right) \otimes I_2 \tag{39.25}$$

We have the extreme case where the parallaxes are perfect: $\sigma_p = 0$:

$$\Sigma_{xx} = \sigma_{x_0}^2 \, \mathbf{1}\mathbf{1}^\mathsf{T} \otimes I_2 \tag{39.26}$$

Then all points are 100% correlated.

39.6.2 The 3D Model

We now want to extend the model to observed directions, namely assuming they are correlated. This extension is non-trivial, why we provide an approximate solution.

The reason is that the basic model (39.22) implicitely assumes the projection centers are coplanar, the viewing directions are parallel, the scene is fronto-parallel, and the image coordinates refer to a perspective model. Then a surface patch is mapped to identical image patches, allowing to use the result of Förstner (1998). As soon as the surface element is observed from different directions, this model does not hold anymore. This not only holds for tilted cameras but also for spherical cameras, where the addition in (39.22) cannot be easily replaced.

We therefore exploit the result of Förstner (1998) by modelling the situation is two steps:

1. In the first step, we assume the surface patch is seen along its normal, however, allowing the distance of the projection centers may vary. Then the setup of a simultaneous homogeneous least squares matching is possible. The resulting accuracies refer to the image coordinates ($\Delta \boldsymbol{p}_k$) refer to the scene, and, using the distances s_k to the projection centers can be transformed into individual directional uncertainties, which, due to the isotropy assumption, lead to isotropic directional uncertainties.
2. In the second step, we assume the directional accuracy approximately transfers to directions not being parallel to the normal. This is a valuable approximation if the deviation from the normal is not too large, since the deviation increases with $1/\cos(\alpha_k)$, where α_k is the angle between the observed direction and the normal of the surface patch. Neglecting this factor simulates the situation where the scene is assumed to consist of small spheres, whose relative direction is determined by least squares matching, which is an unlikely but not invalid assumption.

39.6.2.1 Observed Directions parallel to the Normal of a Surface Patch

The result of the previous subsection can directly be used for expressing the lateral uncertainty of the spatial deviations across the direction. Using (39.21) we find the directional uncertainty from

$$\sigma_{\delta_k} = \frac{\sigma_{d_k}}{s_k} \tag{39.27}$$

where the standard deviations σ_{d_k} correspond to the σ_{x_k} in the left bracket of (39.25). Hence we assume the directional errors $\underline{\boldsymbol{d}} = [\underline{\mathbf{d}}_k]$]are isotropic with

$$\Sigma_{dd} = \sigma_{x_0}^2 \, J_K + \sigma_p^2 \, (I_K - J_K/K) \tag{39.28}$$

Since we need the factors

$$w_{d_k=} = \frac{1}{\sigma_{d_k}^2} \tag{39.29}$$

in the weight matrices, which now are not independent we use the weight matrix

$$W_{dd} = \Sigma_{dd}^{-1} = [w_{kk'}] = \frac{1}{\sigma_p^2} I_K - \frac{K\sigma_{x_0}^2 - \sigma_p^2}{K^2 \sigma_p^2 \sigma_{x_0}^2} J_K \tag{39.30}$$

Since the individual weight matrix (39.17) for one direction can be written as

$$\mathsf{W}_k = \mathsf{S}(\mathbf{d}_k)\ (w_{d_k}\mathsf{I}_3)\mathsf{S}^\mathsf{T}(\mathbf{d}_k) \tag{39.31}$$

we obtain the full weight matrix as

$$\mathsf{W} = [\mathsf{W}_{kk'}] = \mathrm{Diag}(\mathsf{S}(\mathbf{d}_k))\,[w_{kk'}\mathsf{I}_3]\,\mathrm{Diag}^\mathsf{T}(\mathsf{S}(\mathbf{d}_k)) \tag{39.32}$$

or more explicit

$$\mathsf{W}_{kk'} = w_{kk'}\mathsf{S}(\mathbf{d}_k)\mathsf{S}(\mathbf{d}_{k'}) \tag{39.33}$$

Therefore, the solution for the 3D point reads as

$$\widehat{\boldsymbol{X}} = \mathsf{N}^{-1}\boldsymbol{n} \quad \text{with} \quad \mathsf{N} = [N_{ij}] = \sum_{k,k'} \mathsf{W}_{kk'} \quad \text{and} \quad \boldsymbol{n} = [n_j] = \sum_{k,k'} \mathsf{W}_{kk'}\boldsymbol{Z}_{k'}\,. \tag{39.34}$$

39.7 Uncertainty of binocular triangulation with omnidirectional cameras

Given is the configuration

- Distance D
- Basis B
- Angular range α
- Effective image diameter/width W
- Matching accuracy σ_δ referring to the direction

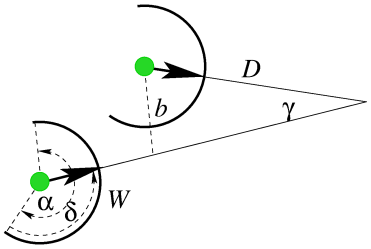


Figure 39.3: Configuration

Then we have for small δ

- The parallactic angle

$$\gamma = \frac{b}{D} \quad \text{or} \quad D\gamma = b \quad \text{and} \quad \mathrm{d}D\,\gamma + D\,\mathrm{d}\gamma = 0 \quad \text{and} \quad \frac{\sigma_\gamma}{\gamma} = \frac{\sigma_D}{D} \tag{39.35}$$

  thus

$$\sigma_D = \frac{D}{\gamma}\sigma_\gamma = \frac{D^2}{b}\sigma_\gamma \tag{39.36}$$

- The pixel size corresponding to direction elements $\Delta\delta$ in [rad] is

$$\Delta\delta = \frac{\alpha}{W} \tag{39.37}$$

  assuming a pixel distance corresponds to the same directional difference, which is an approximation.

- the uncertainty of the measured parallactic angle, as difference of two directions

$$\sigma_\gamma = \sqrt{2}\,\sigma_\delta \tag{39.38}$$

Hence, we finally have the distance accuracy

$$\sigma_D = \sqrt{2}\frac{D}{\gamma}\sigma_\delta \tag{39.39}$$

Since we usually describe the matching accuracy, i.e., the accuracy σ_p of the parallax in pixels, we need to take the resolution into account. Then we have

$$\sigma_\delta = \frac{\alpha}{W}\,\frac{\sigma_p}{\sqrt{2}} \tag{39.40}$$

Then we obtain for the distance

$$\boxed{\sigma_D = \frac{D}{\gamma}\,\frac{\alpha}{W}\,\sigma_p = \frac{D^2}{b}\,\frac{\alpha}{W}\,\sigma_p} \tag{39.41}$$

If we refer to the inverse depth

$$s = \frac{1}{D} \quad \text{with} \quad sD = 1 \quad \text{and} \quad \mathrm{d}s\,D + s\,\mathrm{d}D = 0 \quad \text{and} \quad \frac{\sigma_s}{s} = \frac{\sigma_D}{D} \tag{39.42}$$

we obtain

$$\sigma_s = \frac{s}{D}\sigma_D = \frac{s}{D}\,\frac{D^2}{b}\,\frac{\alpha}{W}\,\sigma_p = \frac{1}{b}\,\frac{\alpha}{W}\,\sigma_p \tag{39.43}$$

from which we may derive the matching accuracy

$$\sigma_p = b\,\frac{W}{\alpha}\,\sigma_s \tag{39.44}$$

if we know the camera, i.e., the viewing angle α, how its image is used (possibly reduced in resolution), i.e., the diameter of the image in pixels, and how large the effective baseline b is.

Part VII

Technical Notes on Mathematics

40 Rational Rotation Matrices

When providing exercises with rotation matrices we may want the elements only contain rational numbers. This can be achieved by using a generalized version of Pythagoras' theorem for quintuples and apply it to the generation of integer quaternions.

40.1 Preface

The note (2021) shows how to generate rotation matrices with rational numbers. This might be useful when generating exercises for students. It was motivated by developing explicit expressions for squares and square roots of quaternions, which correspond to rotations with double and half the angle of a given rotation matrix.

40.2 The problem

When providing exercises with rotation matrices we may want the elements only contain rational numbers, which can be realized using rotations defined by a unit quaternion containing only rational numbers:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad \text{with} \quad |\mathbf{q}| = 1 \quad \text{and} \quad q_i \in \mathbb{Q}\,. \tag{40.1}$$

This is equivalent to use Pythagorean quintuplets $p = (p_0, p_1, p_2, p_3, p_4)$

$$p_0^2 + {p_1}^2 + {p_2}^2 + {p_3}^2 = p_4^2 \tag{40.2}$$

since then

$$q_i = \frac{p_i}{p_4} \tag{40.3}$$

yield a rational unit quaternion $\mathbf{q} = (q, \boldsymbol{q})$ and the corresponding rotation matrix is given by rational elements

$$R = I_3 + 2(qS(\boldsymbol{q}) + S^2(\boldsymbol{q})) \quad \text{with} \quad S(\boldsymbol{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}, \tag{40.4}$$

Usually, the tuples are given in non-decreasing order. As an example, we have the following quintuples

$$p_1(1,1,1,1,2)\,, \quad p_2(4,5,12,16,21) \quad p_3(3,5,11,13,18)\,. \tag{40.5}$$

leading to the rotation matrices

$$R_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad R_2 = \frac{1}{21^2}\begin{bmatrix} -359 & -8 & 256 \\ 248 & -121 & 344 \\ 64 & 424 & 103 \end{bmatrix}, \quad R_3 = \frac{1}{9^2}\begin{bmatrix} -64 & 8 & 49 \\ 47 & -16 & 64 \\ 16 & 79 & 8 \end{bmatrix}. \tag{40.6}$$

The first quintuple p_1 is the one with the smallest elements, the second p_2 contains distinct elements, and the last one p_3 contains primes as the first four elements.

We have the following simple observations:

- Not all five elements can be odd.
- Multiplication with an arbitrary integer does not change the property.
- There exist sequences $(a_1, a_2, ..., a_n, ...)$ of numbers such that for arbitary n we have

$$\sum_{i=1}^{n-1} a_i^2 = a_n^2 \tag{40.7}$$

e.g.,

$$(3, 4, 12, 84, 132, 123254, ...)\,, \tag{40.8}$$

see https://math.stackexchange.com/questions/1632281/how-to-find-pythagorean-triples-and-n-tuples

The problem is to construct such quintuplets. It has been addressed in several ways:

- A list of such quintuples can be found in http://www.techborder.com/education/math/IntMath/Volume_B/Chapter_6/PythagQuints.txt. The list contains two lists, one with maximal elements up to 33, one with maximal elements between 2107 ans 2143. No information is given, on how these quintuplets have been generated.
- Katkar (2019) discussed extensions Pythagoras' theorem, and gives rules how to generate Pythagorean n-tuples ($n > 3$.http://publications.azimpremjifoundation.org/2127/1/2_Extensions%20of%20the%20theorem%20of%20Pythagoras.pdf
- In blog https://groups.google.com/g/sci.math/c/Ujn6cin7mt0 a method is proposed to generate quintuples
- Koecher and Remmert (1991, Chapt. 7) give an explicit algorithm how to generate rational unit quaternions and indicate, this can be used to generate Pythagorean quintuples.

40.3 Generating Pythagorean quintuplets

We start with the construction of rational unit quaternions given by Koecher and Remmert (1991, Chapt. 7). They write:

Every rational quaternion $\mathbf{q} = (\alpha, \boldsymbol{\beta}) \in S^3 \setminus 1, \alpha, \beta_\nu \in \mathbb{Q}$ has the form

$$\alpha = \frac{1 - |\boldsymbol{q}|^2}{1 + |\boldsymbol{q}|^2}\,, \quad \boldsymbol{\beta} = \frac{2\boldsymbol{q}}{1 + |\boldsymbol{q}|^2} \quad \text{with} \quad \boldsymbol{q} := \frac{\boldsymbol{\beta}}{1 + \alpha} \in \mathbb{Q}^3\,, \tag{40.9}$$

Since, for arbitrary $\boldsymbol{q}$ we have:

$$|\mathbf{q}| = \alpha^2 + |\boldsymbol{q}|^2 = \frac{1 - 2|\boldsymbol{q}|^2 + |\boldsymbol{q}|^4}{1 + 2|\boldsymbol{q}|^2 + |\boldsymbol{q}|^4} + \frac{4|\boldsymbol{q}|^2}{1 + 2|\boldsymbol{q}|^2 + |\boldsymbol{q}|^4} = 1\,. \tag{40.10}$$

Starting from four arbitrary values $(a, \boldsymbol{b}) \in \mathbb{Q}^4$ we can generate the quaternion

$$\mathbf{q} = \left(\frac{1 - |\boldsymbol{q}|^2}{1 + |\boldsymbol{q}|^2}, \frac{2\boldsymbol{q}}{1 + |\boldsymbol{q}|^2}\right) \quad \text{with} \quad \boldsymbol{q} = \frac{\boldsymbol{b}}{1 + a}\,. \tag{40.11}$$

Remark: For a scalar b, thus a scalar $q = \tan\frac{\theta}{2}$ we have

$$\mathbf{q} = \left(\frac{1-\tan^2\frac{\theta}{2}}{1+\tan^2\frac{\theta}{2}}, \frac{2\tan\frac{\theta}{2}}{1+\tan^2\frac{\theta}{2}} \right) = (\cos\theta, \sin\theta)\,. \tag{40.12}$$

◇

Therefore, assuming an arbitrary 4-tuple $(a, \boldsymbol{b}) \in \mathbb{Z}^4 \setminus -1$ we can generate a unit quaternion as ratio by expanding (40.11)

$$\mathbf{q} = \left(\frac{1-\frac{|\boldsymbol{b}|^2}{(1+a)^2}}{1+\frac{|\boldsymbol{b}|^2}{(1+a)^2}}, \frac{2\frac{\boldsymbol{b}}{1+a}}{1+\frac{|\boldsymbol{b}|^2}{(1+a)^2}} \right) = \left(\frac{(1+a)^2-|\boldsymbol{b}|^2}{(1+a)^2+|\boldsymbol{b}|^2}, \frac{2(1+a)\boldsymbol{b}}{(1+a)^2-|\boldsymbol{b}|^2} \right) \tag{40.13}$$

$$\mathbf{q} = \frac{\mathbf{x}}{y} \tag{40.14}$$

with integral numerator

$$\mathbf{x} = \mathrm{num}(\mathbf{q}) = \left((1+a)^2 - |\boldsymbol{b}|^2\,, 2(1+a)\boldsymbol{b} \right) \tag{40.15}$$

and integral denominator

$$y = \mathrm{denum}(\mathbf{q}) = (1+a)^2 + |\boldsymbol{b}|^2 \tag{40.16}$$

Observe, we have

$$((1+a)^2 - |\boldsymbol{b}|^2)^2 + 4(1+a)^2|\boldsymbol{b}|^2 = ((1+a)^2 + |\boldsymbol{b}|^2)^2 \tag{40.17}$$

Hence we obtain the quintuplet

$$\boxed{p = \left(u^2 - |\boldsymbol{v}|^2, 2u\boldsymbol{v}, u^2 + |\boldsymbol{v}|^2\right) \quad \text{with} \quad (u, \boldsymbol{v}) \in \mathbb{Z}^4\,.} \tag{40.18}$$

The denominator of the resulting rotation matrix can be shown to be $(u^2 + |\boldsymbol{v}|^2)^2$ at maximum.

Remark: This can easily be proven using Euler's four-square theorem. It results from the norm of the square of a quaternion:

$$|\mathbf{q}^2|^2 = |\mathbf{q}|^2 \cdot |\mathbf{q}|^2 = |\mathbf{q}|^4 \tag{40.19}$$

which holds, since for any two quaternions $\mathbf{x}$ and $\mathbf{y}$ we have $|\mathbf{xy}| = |\mathbf{x}| \cdot |\mathbf{y}|$. If we choose

$$\mathbf{q} = (u, \boldsymbol{v}) \tag{40.20}$$

we obtain (40.18). ◇

Remark: Eq. (40.18) works in all dimensions $d \geq 1$ for $\boldsymbol{v} \in \mathbb{Z}^n$. For $d = 1$ we obtain the classical formula for a Pythagorean triplet (x, y, z)

$$(u^2 - v^2, 2uv, u^2 + v^2) \quad \text{with} \quad u, v \in \mathbb{N} \quad \text{and} \quad u > v\,. \tag{40.21}$$

◇

Example 40.3.30: ***Rotation from quatrupel.*** We randomly generate a quadruple $\boldsymbol{a} = (u, \boldsymbol{v}) \in \mathbb{Z}^4$, derive the Pythagorean quintuple $\boldsymbol{p}$, the rational quaternion $\mathbf{q}$, and finally the rational rotation matrix. As an example, we obtain the sequence

$$\boldsymbol{a} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ -1 \end{bmatrix}, \quad \boldsymbol{p}(\boldsymbol{a}) = \begin{bmatrix} -4 \\ 4 \\ 0 \\ -2 \\ 6 \end{bmatrix}, \quad \mathbf{q}(\boldsymbol{p}) = \frac{1}{3}\begin{bmatrix} -2 \\ 2 \\ 0 \\ -1 \end{bmatrix}, \quad R(\mathbf{q}) = \frac{1}{25}\begin{bmatrix} 15 & 0 & -20 \\ 16 & 15 & 12 \\ 12 & -20 & 9 \end{bmatrix}, \tag{40.22}$$

thus a rotation matrix with rational elements. ◇

41 On the Cayley Transform

We address the Cayley transform relevant for representing rotations and present an excerpt of his paper (1946)

41.1 Preface

The note (2020) addresses the Cayley transform, and contains an excerpt of the paper by Cayley (1846).

41.2 Summary

The Cayley transformation $\lambda(A)$ of a matrix A not having eigenvalue -1 is given by

$$\begin{aligned} B = \lambda(A) &:= (I - A)(I + A)^{-1} && (41.1)\\ &= (I + A)^{-1}(I - A)\,. && (41.2) \end{aligned}$$

Its inverse is

$$\begin{aligned} A = \lambda^{-1}(A) &:= (I - B)(I + B)^{-1} && (41.3)\\ &= (I + B)^{-1}(I - B) = \lambda(B)\,. && (41.4) \end{aligned}$$

Hence it is an involutory transformation as $A = \lambda(\lambda(A))$. Generally, the factors can be exchanged.

There exists an alternative definition. The forward transformation for matrices not having eigenvalue $+1$ is

$$\begin{aligned} B = c(A) &:= (I + A)(I - A)^{-1} && (41.5)\\ &= (I - A)^{-1}(I + A)\,. && (41.6) \end{aligned}$$

The inverse transformation is

$$\begin{aligned} A = c^{-1}(B) &:= (B + I)^{-1}(B - I) && (41.7)\\ &= (B - I)(B + I)^{-1}\,. && (41.8) \end{aligned}$$

Here, the inverse transformation is not identical to the forward transformation.

The forward transformations lead to mutually inverse matrices

$$c(A) = [\lambda(A)]^{-1}\,, \tag{41.9}$$

see the multiplication of (41.1) and (41.5). Similarly, the two inverse transformations are related by

$$c(B) = -\lambda(B)\,, \tag{41.10}$$

see (41.3) and (41.8).

The relevance of the Cayley transformation results from its ability to define rotations. If the matrix A is skew, then B is a rotation matrix. E.g. with the skew matrix

$$S(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{41.11}$$

we have

$$R_\lambda(\boldsymbol{x}) = (I - S(\boldsymbol{x}))(I + S(\boldsymbol{x}))^{-1} \tag{41.12}$$

and

$$R_c(\boldsymbol{x}) = (I + S(\boldsymbol{x}))(I - S(\boldsymbol{x}))^{-1}\,. \tag{41.13}$$

The two rotations are mutually inverse. With the above definition of the skew symmetric matrix the rotation matrix $R_c(\boldsymbol{x})$ is a positive rotation around the axis $\boldsymbol{x}$, whereas the rotation matrix $R_\lambda(\boldsymbol{x})$ is a negative rotation around the axis $\boldsymbol{x}$. This is why the second version $R_c(\boldsymbol{x})$ of the definition is used frequently for representing rotations.

In the original publication Cayley (1846, p. 121) gives the example with

$$A' = \begin{bmatrix} 1 & \nu & -\mu \\ -\nu & 1 & \lambda \\ \mu & -\lambda & 1 \end{bmatrix} \tag{41.14}$$

hence with

$$\boldsymbol{u}' = -\begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix}. \tag{41.15}$$

This is equivalent to using $A = -S(\boldsymbol{u})$ with

$$\boldsymbol{u} = \begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix}. \tag{41.16}$$

He says: the vector $[\lambda,\, \mu,\, \nu]$ can be interpreted as a rotation around axis $\boldsymbol{u}$ with angle θ if

$$\theta = 2\arctan|\boldsymbol{u}| \tag{41.17}$$

or

$$\begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix} = \tan\left(\frac{1}{2}\theta\right)\,\frac{\boldsymbol{u}}{|\boldsymbol{u}|}\,, \tag{41.18}$$

which is consistent with

$$R(\boldsymbol{r},\theta) = R_c(\boldsymbol{u}) = R_\lambda(-\boldsymbol{u}) \quad \text{with} \quad \boldsymbol{r} = \frac{\boldsymbol{u}}{|\boldsymbol{u}|}\,. \tag{41.19}$$

41.3 Comparison

41.3.1 Cayley's derivation

Starting from the definition[1] of a matrix $\boldsymbol{\lambda}$

(C1) $$\lambda_{rs} = -\lambda_{sr}\,, \text{ for } r \neq s \tag{41.20}$$

and

(C2) $$\lambda_{rr} = 1 \tag{41.21}$$

With some skew matrix S this can be written as

$$\boldsymbol{\lambda} := I + S \quad \text{with} \quad \boldsymbol{\lambda}^{\mathsf{T}} := I - S = 2I - (I + S) =: 2\mathbf{1} - \boldsymbol{\lambda}\,. \tag{41.22}$$

The transformation of two points $\boldsymbol{P} = [P_s] := \boldsymbol{p}$ and $\boldsymbol{Q} = [Q_r] := \boldsymbol{q}$ are defined by

(C3) $$\boldsymbol{P} = \boldsymbol{\lambda x} \quad \text{and} \quad \boldsymbol{Q} = \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{x} \tag{41.23}$$

or

$$\boldsymbol{p} = (I + S)\boldsymbol{x} \quad \text{and} \quad \boldsymbol{q} = (I - S)\boldsymbol{x}\,. \tag{41.24}$$

The inverse relation is

(C4) $$K\boldsymbol{x} = \boldsymbol{\Lambda P} \quad \text{and} \quad K\boldsymbol{x} = \boldsymbol{\Lambda}^{\mathsf{T}} \boldsymbol{Q}\,, \tag{41.25}$$

We use the adjoint matrix ($A^* = |A|A^{-1}$)

$$\boldsymbol{\Lambda} = \boldsymbol{\lambda}^* := (I + S)^* \tag{41.26}$$

of $\boldsymbol{\lambda}$ and have

$$|\boldsymbol{\lambda}|\boldsymbol{x} = \boldsymbol{\lambda}^* \boldsymbol{P} \quad \text{and} \quad |\boldsymbol{\lambda}|\boldsymbol{x} = \boldsymbol{\lambda}^{*\mathsf{T}} \boldsymbol{Q}\,. \tag{41.27}$$

This is equivalent to

$$|I + S|\,\boldsymbol{x} = (I + S)^*\,\boldsymbol{p} \quad \text{and} \quad |I + S|\,\boldsymbol{x} = (I - S)^{*\mathsf{T}}\,\boldsymbol{q}\,. \tag{41.28}$$

Therefore we have

(C5) $$\boldsymbol{\lambda\Lambda} = KI \quad \text{or} \quad (I + S)(I + S)^* = |I + S|I\,. \tag{41.29}$$

From (41.25) we see

(C6) $$\boldsymbol{\Lambda P} = \boldsymbol{\Lambda}^{\mathsf{T}} \boldsymbol{Q} \quad \text{or} \quad (I + S)^*\,\boldsymbol{p} = (I - S)^{*\mathsf{T}}\,\boldsymbol{q}\,. \tag{41.30}$$

Premultiplication with $\boldsymbol{\lambda}^{\mathsf{T}}$ yields

(C7) $$\boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{\Lambda P} = \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{\Lambda}^{\mathsf{T}} \boldsymbol{Q} \tag{41.31}$$

or $$(I + S)^{\mathsf{T}} (I + S)^*\,\boldsymbol{p} = (I - S)(I - S)^{*\mathsf{T}}\,\boldsymbol{q}\,. \tag{41.32}$$

Using (41.29) this is identical to

(C8) $$\boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{\Lambda}^{\mathsf{T}} \boldsymbol{Q} = K\boldsymbol{Q} \quad \text{or} \quad (I - S)(I - S)^{*\mathsf{T}}\,\boldsymbol{q} = |I + S|\,\boldsymbol{q}\,. \tag{41.33}$$

With (41.20), (41.21), and (41.22) we have

(C9) $$\boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{\Lambda} = 2\boldsymbol{\Lambda} - \boldsymbol{\lambda\Lambda} \tag{41.34}$$

or $$(I - S)(I + S)^* = 2(I + S)^* - (I + S)(I + S)^*\,. \tag{41.35}$$

[1] We refer to Cayley's equations with (C#).

which with (41.29) is

$$\text{(C9)} \qquad \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\Lambda} = 2\boldsymbol{\Lambda} - K\mathbf{1} \quad \text{or} \quad (I-S)(I+S)^{*} = 2(I+S)^{*} - KI \tag{41.36}$$

The transformation of $\boldsymbol{P}$ yields

$$\text{(C9)} \qquad \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{P} = 2\boldsymbol{\Lambda}\boldsymbol{P} - K\boldsymbol{P} \tag{41.37}$$

$$\text{or} \qquad (I-S)(I+S)^{*}\boldsymbol{p} = 2(I+S)^{*}\boldsymbol{p} - K\boldsymbol{p}\,. \tag{41.38}$$

Using (41.29), (41.30) and (41.37) in (41.31) we obtain

$$\text{(C12)} \qquad K\boldsymbol{Q} = 2\boldsymbol{\Lambda}\boldsymbol{P} - K\boldsymbol{P} \quad \text{or} \quad K\boldsymbol{q} = 2(I+S)^{*}\boldsymbol{p} - \boldsymbol{p} \tag{41.39}$$

and also

$$\text{(C13)} \qquad K\boldsymbol{P} = 2\boldsymbol{\Lambda}\boldsymbol{Q} - K\boldsymbol{Q} \quad \text{or} \quad K\boldsymbol{p} = 2(I+S)^{*}\boldsymbol{q} - \boldsymbol{q} \tag{41.40}$$

Now we set

$$\text{(C14)} \qquad \boldsymbol{\alpha}: \quad \begin{cases} K\alpha_{rs} = 2\boldsymbol{\Lambda}_{rs}\,, \text{ for } r \neq s \\ K\alpha_{rr} = 2\boldsymbol{\Lambda}_{rr} - K \end{cases} \tag{41.41}$$

$$\text{or} \qquad KA = 2(I+S)^{*} - KI\,. \tag{41.42}$$

and therefore simplify (41.39) and (41.40) leading to

$$\text{(C15)} \qquad \boldsymbol{Q} = \boldsymbol{\alpha}\boldsymbol{P} \quad \text{and} \quad \boldsymbol{P} = \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{Q} \quad \text{or} \quad \boldsymbol{q} = A\boldsymbol{p} \quad \text{and} \quad \boldsymbol{p} = A^{\mathsf{T}}\boldsymbol{q}\,. \tag{41.43}$$

Therefore we have

$$\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathsf{T}} = \mathbf{1} \quad \text{or} \quad AA^{\mathsf{T}} = I \tag{41.44}$$

Therefore the matrix $\boldsymbol{\alpha} := A$ is a rotation matrix, rotating $\boldsymbol{p}$ to $\boldsymbol{q}$.

41.3.2 Relation to classical representations

The matrix A in (41.41) can be written as

$$\begin{aligned} A &= (2(I+S)^{*}/|I+S| - I) && (41.45)\\ &= 2(I+S)^{-1} - I && (41.46)\\ &= 2(I+S)^{-1} - (I+S)^{-1}(I+S) && (41.47)\\ &= (I+S)^{-1}(2I - (I+S)) && (41.48)\\ &= (I+S)^{-1}(I-S) = (I-S)(I+S)^{-1} && (41.49) \end{aligned}$$

Deriving S from A uses (41.49). We obtain

$$\begin{aligned} S &= \left[\frac{1}{2}(A+I)\right]^{-1} - I && (41.50)\\ &= 2(A+I)^{-1} - I && (41.51)\\ &= 2(A+I)^{-1} - (A+I)^{-1}(A+I) && (41.52)\\ &= (A+I)^{-1}(2I - (A+I)) && (41.53)\\ &= (A+I)^{-1}(I-A) = (I-A)(A+I)^{-1} && (41.54) \end{aligned}$$

Hence, Cayley's setup yields the involutory transformation.

Observe, the last two derivations hold for arbitrary matrices A and S, as long as the inverses exist.

41.4 Proof of pair of Cayley transformations

We want to prove $A = \lambda(\lambda(A))$. We use (41.46) and (41.49)

$$\lambda(A) = (I - A)(I + A)^{-1} = 2(I + A)^{-1} - I \tag{41.55}$$

Then we obtain

$$\begin{aligned}
\lambda(\lambda(A)) &= \lambda\left(\underbrace{2(I + A)^{-1} - I}_{\lambda(A)}\right) && (41.56)\\
&= 2\left(I + (\underbrace{2(I + A)^{-1} - I}_{\lambda(A)})\right)^{-1} - I && (41.57)\\
&= 2\left((2(I + A)^{-1}\right)^{-1} - I && (41.58)\\
&= 2\left(\frac{1}{2}(I + A)\right) - I && (41.59)\\
&= A && (41.60)
\end{aligned}$$

Part VIII

Miscellaneous Notes

42 Necessary Width of a Corridor to Transport a Piano around a Corner

We analyse under which conditions a piano can be moved around a corner connecting two mutually orthogonal corridors having different widths. In case the corridors have a width between the two sides of the piano, the second corridor needs to be wider if the first corridor is narrower.

42.1 Preface

The note (2011) is motivated by a discussion with a friend who moved to a new flat and had a piano, asking me, how wide the two corridors at a corner need to be in order to be able to move the piano around the corner.

42.2 Problem

Gíven is a piece of furniture, imagine a piano, of depth a and width b which needs to be moved through a corridor of width $a \leq u \leq b$ and then turned at a 90°corner into a second corridor. The question is, how wide the second corridor needs to be.

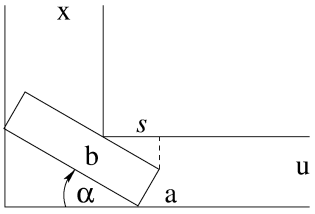


Figure 42.1: Situation: We seek to know the minimum width x of the second corridor, such that the rectangular piece of furniture can be moved around the corner.

42.3 Setup

We determine the width x as a function of the angle α.

We start with the position s of the left part of the piano:

$$u = a\cos\alpha + s\tan\alpha\,, \tag{42.1}$$

from which we can follow

$$s = \frac{u - a\cos\alpha}{\tan\alpha}\,. \tag{42.2}$$

Now we determine the total length $s + x$ of the piano in horizontal direction

$$s + x = a\sin\alpha + b\cos\alpha\,. \tag{42.3}$$

From this we may determine the current width x of the second corridor as a function of α

$$x(\alpha) = a\sin\alpha + b\cos\alpha - \frac{u - a\cos\alpha}{\tan\alpha}\,. \tag{42.4}$$

For three examples this function is shown in the Fig. 42.2

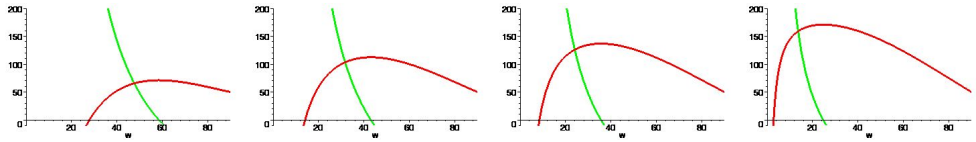


Figure 42.2: Width x in cm (red) as function of the angle α for $b = 200$ cm and $a = 50$ cm for values $u = (150, 100, 80, 60)$ cm. Yellow: The first derivative, whose zero yields the maximum for x

u cm	$\alpha_0{}^\circ$	$x_{\min}$	Figur
150	58.5	71.2	left
100	43.0	112.3	mid-left
80	35.6	136.8	mid right
60	24.7	170.9	right

Table 42.1: Maximal values for the width x for the cases in Fig. 42.2

42.4 Smallest width of the second corridor

The second corridor needs to have a minimum width. It results from the maximum of the function $x(\alpha)$

$$x' = \frac{\mathrm{d}x(\alpha)}{\mathrm{d}\alpha} = 0 \tag{42.5}$$

This derivative is

$$x' = \frac{b\sin(\alpha) - b\sin(\alpha)\left(\cos(\alpha)\right)^2 - u + t\cos(\alpha)}{-1 + \left(\cos(\alpha)\right)^2} \tag{42.6}$$

We only need to take the numerator. Hence, we need to solve the equation

$$0 = b\sin(\alpha) - b\sin(\alpha)\left(\cos(\alpha)\right)^2 - u + a\cos(\alpha) \tag{42.7}$$

for α. This usually yields a value in the range 0 bis 90 °. The minimal value for the width of the second corridor then is

$$x_{\min} = x(\alpha_0)\,. \tag{42.8}$$

Four values are given in the table. We find the intuitive result: The wider the first corridor, the narrower the second corridor can be.

42.5 Solution with envelope curve

We follow an idea of J. Meidow. If we move the piano around a corner, the longer side not touching the wall moves: The envelope of all these straight lines provides the boundary of the region over which the piano is moved, see Fig. 42.3

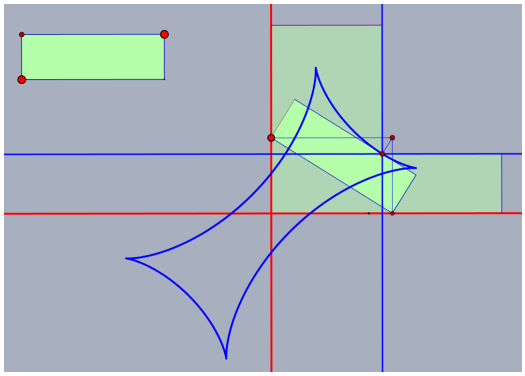


Figure 42.3: Moving a piano around a corner connecting two corridors of different width, figure generated with Conderella

42.5.1 Moving a ladder – the astroid

42.5.1.1 The envelope solution

In case the piano would be a ladder having length b, we could move it along the corner. The envelope then is known to be an asteroid, see https://mathworld.wolfram.com/Astroid.html: Starting from the line through $[0, by(t)]$ and $[bx(t), 0]$ with $x^2(t) + y^2(t) = b^2$, and $x = t$, thus the line through $[0, \sqrt{b^2 - t^2}]$ and $[bt, 0]$ which is

$$\ell : \quad \frac{x}{bt} + \frac{y}{\sqrt{b^2 - t^2}} - 1 = 0 \tag{42.9}$$

or

$$f(t, x, y) = y + \frac{\sqrt{b^2 - t^2}}{t}(x - t) \tag{42.10}$$

with its partial derivative to t

$$f_t(t, x, y) = \frac{t^3 - b^2 x}{t^2\sqrt{b^2 - t^2}} \tag{42.11}$$

we need to eliminate t from

$$f(t, x, y) = \quad \text{and} \quad f_t(t, x, y) = 0\,. \tag{42.12}$$

Hence

$$x = \frac{t^3}{b^2} \quad \text{and} \quad y = -\frac{\sqrt{b^2 - t^2}}{t}(t^3/b^2 - t) = \frac{(b^2 - t^2)^{3/2}}{b^2} \tag{42.13}$$

Since

$$t^2 = b^{4/3}x^{2/3} \tag{42.14}$$

we have

$$y = \frac{(b^2 - b^{4/3}x^{2/3})^{3/2}}{b^2} = \frac{(b^{4/3}(b^{2/3} - x^{2/3}))^{3/2}}{b^2} = (b^{2/3} - x^{2/3})^{3/2} \tag{42.15}$$

or

$$x^{2/3} + y^{2/3} = b^{2/3}\,. \tag{42.16}$$

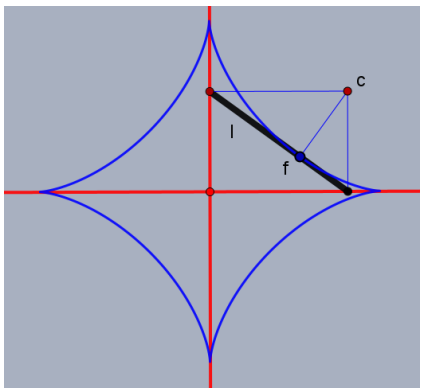


Figure 42.4: Moving ladder

42.5.1.2 A geometric solution

An alternative derivation just follows geometric insight, see Fig 42.4. If the ladder is slightly moving, it rotates around the point $\boldsymbol{c}([t, \sqrt{1-t^2}])$, since the left endpoint moves vertically and the right endpoint moved horizontally. Hence, the point on the ladder which stays at constant distance from this point is the footpoint $\boldsymbol{f}$ of $\boldsymbol{c}$ on the line $\boldsymbol{\ell}$. The line has homogeneous coordinates

$$\mathbf{l} = \begin{bmatrix} \sqrt{b^2 - t^2} \\ t \\ t\sqrt{b^2 - t^2} \end{bmatrix} \tag{42.17}$$

The foot point of the centre

$$\mathbf{c} = \begin{bmatrix} t \\ \sqrt{b^2 - t^2} \\ 1 \end{bmatrix} \tag{42.18}$$

is given by

$$\mathbf{f} = \mathsf{S}(\mathbf{l})\mathsf{S}(\mathbf{c})\mathsf{G}_3\mathbf{l} \quad \text{with} \quad \mathsf{S}(\mathbf{b}) = \frac{\partial \mathbf{a} \times \mathbf{b}}{\partial \mathbf{b}} \quad \text{and} \quad \mathsf{G} = \mathrm{Diag}([1, 1, 0])\,, \tag{42.19}$$

see Förstner and Wrobel (2016, eq. (7.18)). This yields the foot point

$$\boldsymbol{f} = \begin{bmatrix} x \\ y \end{bmatrix}_f = \begin{bmatrix} t^3/b^2 \\ (1-t^2)^{3/2}/b^2 \end{bmatrix} \tag{42.20}$$

as in (42.13).

42.5.2 The geometric solution for the rectangle

We now assume the piano has depth a and length b. Then the line is shifted parallel by a, i.e., shifted by

$$\boldsymbol{s} = b \begin{bmatrix} \sqrt{1-t^2} \\ t \end{bmatrix} \tag{42.21}$$

since the normal vector in (42.17) is already normalized.

Thus we need to apply the shift

$$\mathsf{T} = \begin{bmatrix} 1 & 0 & -a\sqrt{1-t^2} \\ 0 & 1 & -at \\ 0 & 0 & 1 \end{bmatrix} \tag{42.22}$$

Hence we obtain the shifted line

$$\mathbf{l}_p = \mathsf{T}^\circ\mathbf{l} \quad \text{with} \quad \mathsf{T}^\circ = |\mathsf{T}|\mathsf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a\sqrt{1-t^2} & -at & 1 \end{bmatrix} \tag{42.23}$$

yielding

$$\mathbf{l}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a\sqrt{b^2-t^2} & -at & 1 \end{bmatrix} \begin{bmatrix} \sqrt{b^2-t^2} \\ t \\ t\sqrt{b^2-t^2} \end{bmatrix} \tag{42.24}$$

$$= \begin{bmatrix} \sqrt{b^2-t^2} \\ t \\ -a(b^2-t^2) - at^2 + t\sqrt{b^2-t^2} \end{bmatrix} = \begin{bmatrix} \sqrt{b^2-t^2} \\ t \\ -ab^2 + t\sqrt{b^2-t^2} \end{bmatrix} \tag{42.25}$$

Similarly we obtain from $\mathbf{f}_p = \mathsf{S}(\mathbf{l}_p)\mathsf{S}(\mathbf{c})\mathsf{G}_3\mathbf{l}_p$

$$\boldsymbol{f}_p = \begin{bmatrix} t^3/b^2 + a\sqrt{b^2-t^2}/b \\ \left(b^2-t^2\right)^{3/2}/b^2 + a\,t/b \end{bmatrix}. \tag{42.26}$$

For $a = 0$ this specializes to (42.20).

Substituting $t = b\cos\alpha$ into (42.26) confirms the numbers in Tab. 42.1, observing that $u = y$.

43 Line from Four Observed Lines Passing through Equidistant Points

We provide a constructive solution to the following pose estimation problem: A car is driving on an unknown straight path with an unknown, but constant velocity. It is observed from another moving vehicle which knows it's own position (full pose: position and direction) and observes the bearing (direction) to the other car at constant time intervals. The task is to determine the location of the path and velocity of the other car. A geometric solution is given and realized in Cinderella.

43.1 Preface

The note (2005) provides a solution to the following pose estimation problem: A car is driving on an unknown straight path with an unknown, but constant velocity. It is observed from another moving vehicle which knows its on position (full pose: position and direction) and observes the bearing (direction) to the other car at constant time intervals. The task is to determine the location of the path and velocity of the other car. A geometric solution is given and realized in Cinderella

43.2 Problem

Given are four lines $\mathbf{l}_i, i = 1, 2, 3, 4$, the viewing lines of the moving vehicle to the car. Find line $\mathbf{m}$ such that distances of intersection points $\mathbf{x}_i = \mathbf{l}_i \wedge \mathbf{m}$ are the same.

Example 43.2.31: ***Sensitivity of the estimated path of a moving vehicle.*** The Fig. 43.1 shows an example. The poses (A, B, C, L) and the directions (a, b, c, g) of a moving vehicle are given, the four points (X, W, Y, T) of an unknown car, driving on a straight path with constant velocity. The figure shows two situations which only differ in the direction of a from A: Already a small change of the direction may cause a large change in the inferred path of the car. The figures are generated with the geometry package Cinderella (https://www.cinderella.de/tiki-index.php). ◇

Remark: I was not aware of previous publications, e.g., geometric solution: https://ricojansen.nl/downloads/the_four_bearings_method_v2,Kuikueg.pdf, algebraic solution: https://www.mathscinotes.com/2015/04/computing-a-ships-course-from-four-bearings/, video: https://www.youtube.com/watch?v=z2nr2G2SRGY ◇

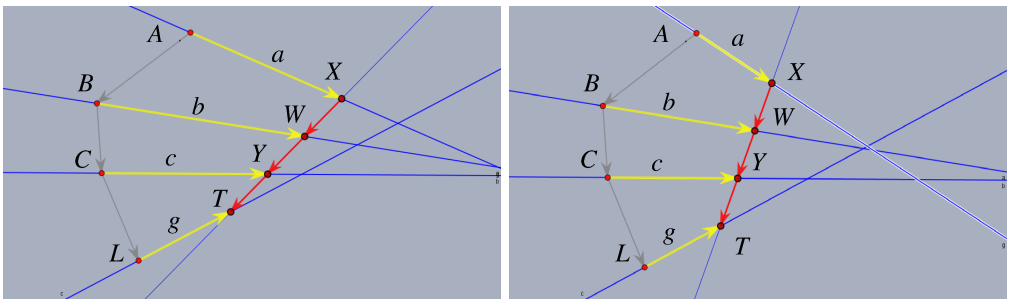


Figure 43.1: Four consecutive points (X, W, Y, T) on an unknown line with equal distances are observed from known positions (A, B, C, L) with known rays (a, b, c, g). Slightly changed direction a leads to quite different path of car.

43.3 Formalization

We assume

$$\mathbf{m} = \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{m}_h \\ m_0 \end{bmatrix} ,$$

thus the line should not go through the origin.

The intersection points are

$$\mathbf{x}_i = \mathbf{l}_i \times \mathbf{m} = \mathsf{S}(\mathbf{l}_i)\mathbf{m} = \mathsf{S}(\mathbf{m})\mathbf{l}_i = \begin{bmatrix} \cdot \\ \cdot \\ l_{i1}b - l_{i2}a \end{bmatrix} .$$

Observe

$$x_{ih} = \boldsymbol{m}_h^\mathsf{T} \boldsymbol{l}_{ih}^\perp .$$

The double area of the triangle ijO is

$$2A_{ij} = h\, d_{ij} = \frac{|\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{e}_3|}{x_{ih} x_{jh}}$$

and proportional to the distance d_{ij} Observe in this special case due to $m_0 = 1$

$$|\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{e}_3| = |\mathbf{m} \times \mathbf{l}_i, \mathbf{m} \times \mathbf{l}_j, \boldsymbol{e}_3| = |\mathbf{l}_i, \mathbf{l}_j, \mathbf{m}| ,$$

thus

$$2A_{ij} = \frac{|\mathbf{l}_i, \mathbf{l}_j, \mathbf{m}|}{\boldsymbol{m}_h^\mathsf{T}\, \boldsymbol{l}_{ih}^\perp\, \boldsymbol{l}_{jh}^{\perp\mathsf{T}}\, \boldsymbol{m}_h}$$

Therefore, we have

$$2A_{jk} = -\frac{|\mathbf{l}_k, \mathbf{l}_j, \mathbf{m}|}{\boldsymbol{m}_h^\mathsf{T}\, \boldsymbol{l}_{jh}^\perp\, \boldsymbol{l}_{kh}^{\perp\mathsf{T}}\, \boldsymbol{m}_h} .$$

For three equidistant points the area difference $= 0$

$$\begin{aligned} 0 = (2A_{ij} - 2A_{jk})(x_{ih} x_{jh} x_{kh}) &= |\mathbf{l}_i, \mathbf{l}_j, \boldsymbol{m}| x_{kh} + |\mathbf{l}_k, \mathbf{l}_j, \boldsymbol{m}| x_{ih} && (43.1) \\ &= |\mathbf{l}_i x_{kh} + \mathbf{l}_k x_{ih}, \mathbf{l}_j, \boldsymbol{m}| && (43.2) \\ &= \left| \mathbf{l}_i\, \boldsymbol{m}_h^\mathsf{T} \boldsymbol{l}_{kh}^\perp + \mathbf{l}_k\, \boldsymbol{m}_h^\mathsf{T} \boldsymbol{l}_{ih}^\perp, \mathbf{l}_j, \begin{bmatrix} \boldsymbol{m}_h \\ 1 \end{bmatrix} \right| . && (43.3) \end{aligned}$$

This expression is a quadratic and homogeneous polynomial

$$\boldsymbol{m}_h^\mathsf{T} K \boldsymbol{m}_h + \boldsymbol{k}^\mathsf{T} \boldsymbol{m}_h = k_{02}a^2 + k_{11}ab + k_{20}b^2 + k_{01}a + k_{10}b = 0$$

in (a, b), as when developing the determinant w.r.t. the last column we never obtain expressions which are independent on a or b. Thus, $a = b = 0$ always is a solution.

Moreover, this constraint it is valid either if $\mathbf{m} = [a, b, 1]^\mathsf{T} = \lambda[l_{j1}, l_{j2}, l_{j3}]^\mathsf{T} = \boldsymbol{l}_j$ or especially if

$$a = \frac{l_{j1}}{l_{j2}} b \; .$$

We now need two constraints for determining a and b. We may use the two quadratic homogeneous polynomials

$$0 = A_{23} - A_{12} \propto P(a, b) \tag{43.4}$$

and

$$0 = A_{34} - A_{23} \propto Q(a, b) \; . \tag{43.5}$$

These have four solutions. In case we solve $P(a, b) = 0$ for b and substitute the solution into $Q(a, b) = 0$ we obtain a fourth order polynomial $R(a)$ in a, which is of the form

$$R(a) \propto aS(a) \; ,$$

where $S(a)$ is a third order polynomial. This is, because $a = b = 0$ always is a solution.

We know we have another two solutions

$$a_2 = \frac{l_{21}}{l_{22}} b_2$$

and

$$a_3 = \frac{l_{31}}{l_{32}} b_3 \; ,$$

as for any b and any ratio a/b we can solve P and Q for a.

Thus, the fourth solution, which is the only non-trivial one, results from the polynomial division

$$a - a_4 \propto S(a) : [(a - a_2)(a - a_3)] \; ,$$

as

$$S(a) = k(a - a_2)(a - a_3)(a - a_4)$$

The general solution in Maple already contains this explicitly.

The polynomial division can be realized in Matlab, e.g., we have `deconv([1,3,3,1],[1,1])` `=> [1,2,1]` as

$$\frac{(x+1)^3}{x+1} = \frac{x^3 + 3x^3 + 3x + 1}{x+1} = x^2 + 2x + 1 = (x+1)^2$$

.

43.4 Construction

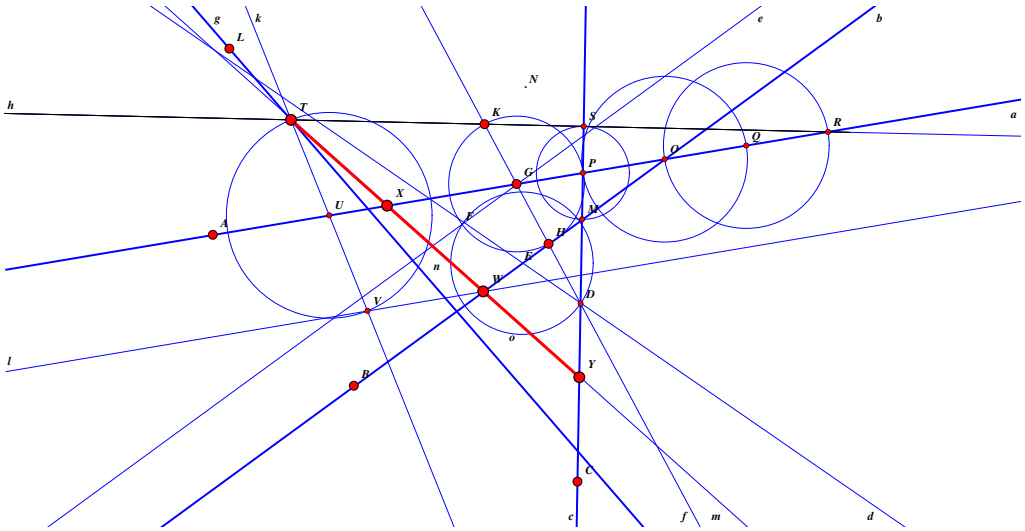
Given Lines $(A, a), (B, b), (C, c), (L, g)$.[1] For arbitrary $D \in c$ and $H \in b$ we obtain line f such that $H = f \cap b$ and $G = f \cap a$ and $DH = HG$ (via intersection of parallel to b at equal distance to b as point D to b [2]).

Then one observes (by interactive construction using CINDERELLA): the prolongation point K at equal distance to G runs on a straight line h when changing reference point D on c. Thus the intersection point T of h and g[3] sits on the sought trajectory. Thus the quadruple (D, H, G, K) corresponds to the quadruple (Y, W, X, T). In case T would be known, then we could construct the triple (T, X, W) starting from T in the same way as we did construct the triple (D, H, G).

For doing this we need to be able to construct the line h.

We observe (without proof):

[1] in the original note: "$(A, a), (C, c), (D, d), (L, k)$"
[2] in the original note "C to b"
[3] in the original note "k"



- The three points $M = c \cap b$, $P = c \cap a$ and $S = c \cap h$ on c are equidistant. This allows to construct S.
- The three points $P = a \cap c$, $O = a \cap b$ and $R = a \cap h$ have distance ratio 1:2. This allows to construct R.
- The line h is the line joining S and R.

44 Peeling a Mandarin or the Spherical Archimedean Spiral

We discuss the question which form a planarized mandarin peel has, in case it is peeled with a constant width, thus along a spherical Archimedean spiral. We formalize the problem and develop a scheme to visualize a planarized Archimedean spiral.

44.1 Preface

44.2 The task

We describe the approximate flattening of a sphere using an Archimedean spiral, see Fig. 44.1

44.3 The Archimedean spiral in the plane

The archimedian spiral in the plane is defined in polar coordinates as

$$r = bt \tag{44.1}$$

where the parameter b describes the velocity of the change in distance to the origin. In cartesian coordinates we obtain the parametric form

$$x = r\cos t = bt\cos t\,, \tag{44.2}$$
$$y = r\sin t = bt\sin t\,. \tag{44.3}$$

Fig. 44.2 shows four branches of length 2π of a spiral with $b = 0.5/(2\pi)$ and radial rays, partitioning the area between neighbouring branches in triangles and quadrangles.

The arclength results from the differentials

$$\dot{x} := \frac{\partial x}{\partial t} = b(\cos t - t\sin t) \tag{44.4}$$
$$\dot{y} := \frac{\partial y}{\partial t} = b(\sin t + t\cos t) \tag{44.5}$$

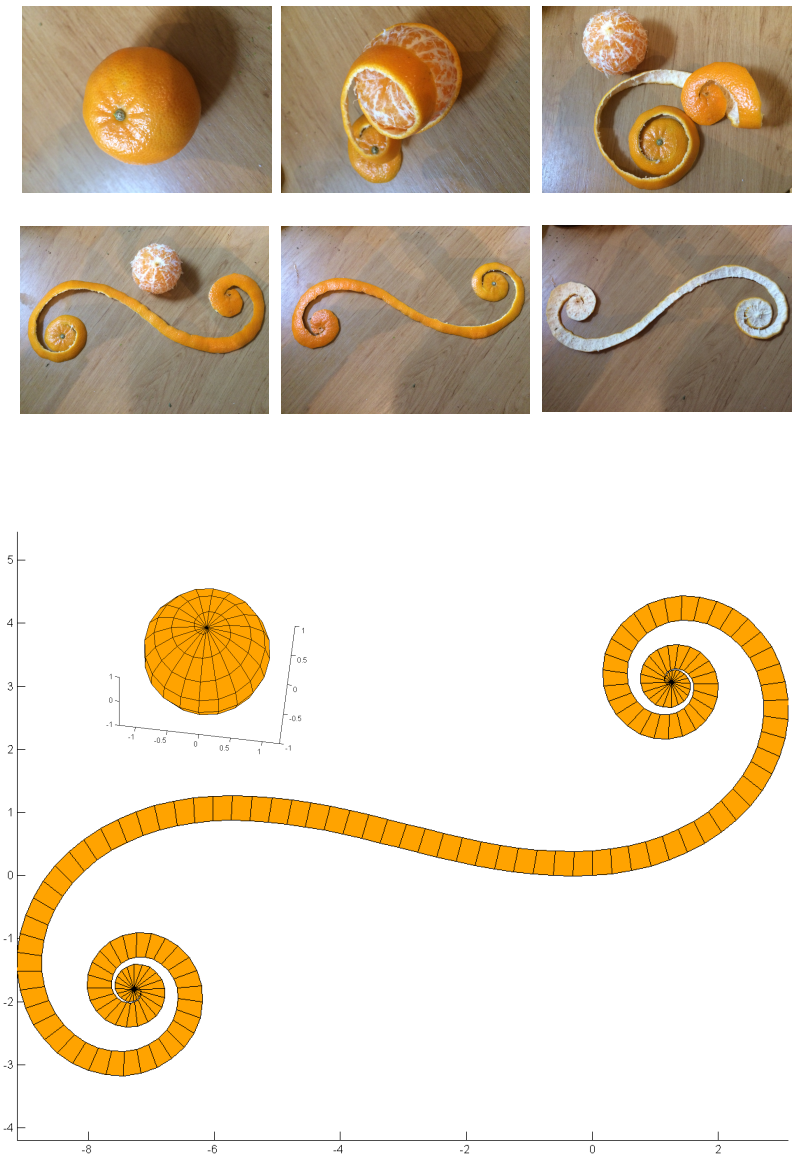


Figure 44.1: Peeling a mandarin leading to a double spiral when planarized (above) and its mathematical simulation

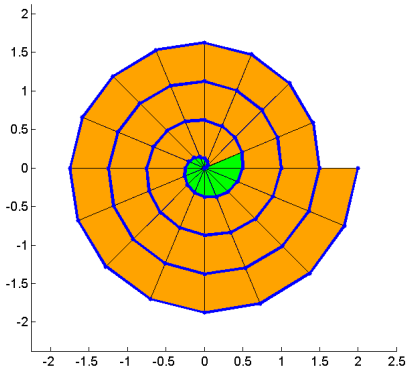


Figure 44.2: Archimedean spiral with $b = 0.5/(2\pi)$, having four branches. The area between the branches is partitioned into triangles (close to the centre) and quadrangles

hence the length element is

$$\mathrm{d}s = \sqrt{\dot{x}^2 + \dot{y}^2}\mathrm{d}t = b\sqrt{1+t^2}\,\mathrm{d}t\,. \tag{44.6}$$

For deriving the curvature, we need the second derivatives:

$$\ddot{x} \;:=\; \frac{\partial \dot{x}}{\partial t} = -b\,(2\,\sin(t) + t\,\cos(t)) \tag{44.7}$$

$$\ddot{y} \;:=\; \frac{\partial \dot{y}}{\partial t} = +b\,(2\,\cos(t) - t\,\sin(t))\;. \tag{44.8}$$

The curvature then is given by

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2+\dot{y}^2)^{3/2}} = \frac{1}{b}\frac{(t^2+2)}{(t^2+1)^{3/2}}\,, \tag{44.9}$$

which obviously decreases with increasing t, namely $\kappa \approx 1/(bt)$ for large t.

44.4 The Archimedean spiral on the unit sphere

44.4.1 Starting at the north pole

The (Archimedean) spiral on the sphere may start at the north pole, surround the sphere several times and arrive at the south pole. Using geographical coordinates (λ, ϕ) we have with the parameter b indicating the velocity of the change of the latitude and assuming the spiral leaves the north pole with increasing λ:

$$\phi_1 = \pi/2 - b\lambda \quad \text{with} \quad \lambda \geq 0\,. \tag{44.10}$$

The spiral arrives at the south pole $\phi_1 = -\pi/2$ for

$$\lambda_{S1} = \frac{\pi}{b}\,, \tag{44.11}$$

and therefore passes the equator at

$$\lambda_{E1} = \frac{\pi}{2b}\,. \tag{44.12}$$

Transforming the geographic into Cartesian coordinates we arrive at using $t = \lambda$

$$\boldsymbol{x}(t) = \begin{bmatrix} \cos t \sin(b\,t) \\ \sin t \sin(b\,t) \\ \cos(b\,t) \end{bmatrix}\,. \tag{44.13}$$

The first derivative w.r.t. is

$$\dot{\boldsymbol{x}}(t) = \begin{pmatrix} b\cos(bt)\cos(t) - \sin(bt)\sin(t) \\ \sin(bt)\cos(t) + b\cos(bt)\sin(t) \\ -b\sin(bt) \end{pmatrix} \tag{44.14}$$

$$\kappa_a' = \frac{\cos(bt)\left(2b^2 + \sin(bt)^2\right)}{\left(b^2 + \sin(bt)^2\right)^{3/2}} \tag{44.15}$$

or for small t we obtain

$$\kappa_a' \approx \frac{t^2+2}{b\,(t^2+1)^{3/2}} \tag{44.16}$$

which is consistent with the curvature of the planar spiral.

44.4.2 Starting at the equator

In order to simplify the relations we use the angular variables $(t, r) := (\lambda, \phi)$ and start the spiral at the equator at $t = 0$ and let it converge to the north pole, leading to

$$\boxed{r = bt \quad \text{with} \quad t \in [-\pi/(2b), +\pi/(2b)]\,,} \tag{44.17}$$

which shows the similarity to the Archimedean spiral in the plane. This spiral reaches the north pole $r = \pi/2$ at

$$t_N = \frac{\pi}{2\,b}\,, \tag{44.18}$$

why we limit the range in (44.17).

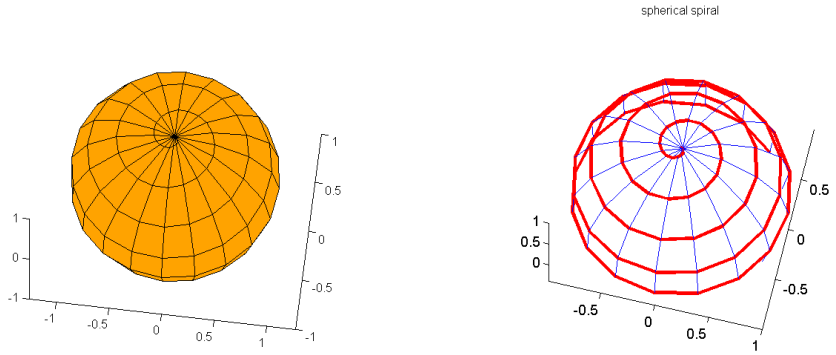


Figure 44.3: Archimedean spiral on the sphere with k branches per hemisphere and $b = 1/(4 * k)$

The representation in Cartesian coordinates is

$$\boldsymbol{x} = \begin{bmatrix} \cos(bt)\cos t \\ \cos(bt)\sin t \\ \sin(bt) \end{bmatrix}. \tag{44.19}$$

Its derivatives are

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -b\sin(bt)\cos t - \cos(bt)\sin t \\ -b\sin(bt)\sin t + \cos(bt)\cos t \\ +b\cos(bt) \end{bmatrix}. \tag{44.20}$$

the local velocity is

$$\boxed{v(t) = \dot{\boldsymbol{x}}(t) = \left|\frac{\partial \boldsymbol{x}}{\partial t}\right| = \sqrt{b^2 + \cos^2(bt)}} \tag{44.21}$$

The second derivatives are

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} 2\,b\,\sin(b\,t)\,\sin(t) - \cos(b\,t)\,\cos(t) - b^2\,\cos(b\,t)\,\cos(t) \\ -\cos(b\,t)\,\sin(t) - 2\,b\,\sin(b\,t)\,\cos(t) - b^2\,\cos(b\,t)\,\sin(t) \\ -b^2\,\sin(b\,t) \end{bmatrix} \tag{44.22}$$

The total curvature is

$$\begin{aligned} \kappa &= \frac{|\dot{\boldsymbol{x}} \times \ddot{\boldsymbol{x}}|}{|\dot{\boldsymbol{x}}|^3} && (44.23)\\ &= \frac{b^6 - b^4\cos(b\,t)^2 + 4\,b^4 - b^2\cos(b\,t)^4 + 4\,b^2\cos(b\,t)^2 + \cos(b\,t)^4}{(b^2+\cos^2(bt))^{3/2}} && (44.24)\\ &= \frac{b^4(b^2+4) + b^2\cos(b\,t)^2(-b^2+4) + \cos(b\,t)^4(-b^2+1)}{(b^2+\cos^2(bt))^{3/2}} && (44.25) \end{aligned}$$

However, we need the curvature of the curve projected onto the tangent plane, see next paragraph.

The torsion is not zero, but

$$\tau(t) = \frac{|[\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}]|}{|\dot{\boldsymbol{x}} \times \ddot{\boldsymbol{x}}|} = \frac{\sin(b\,t)\,\left(2\,b^2 + \cos(b\,t)^2\right)}{b^6 + b^4\sin(b\,t)^2 + 3\,b^4 - b^2\sin(b\,t)^4 - 2\,b^2\sin(b\,t)^2 + 3\,b^2 + \sin(b\,t)^4 - 2\sin(b\,t)^2 + 1} \tag{44.26}$$

For $t = \pi/(2b)$ the maximum is reached with

$$\tau_{\max} = \frac{2}{b^2(b^2+4)}\,. \tag{44.27}$$

44.5 Projection on the tangent frame

We now project the curve onto the tangent frame, with x-coordinate parallel to $\dot{\boldsymbol{x}}$, z-coordinate $\boldsymbol{x}$, and y-coordinate $\boldsymbol{x} \times \dot{\boldsymbol{x}}$

Hence

$$\begin{aligned} {}^t\boldsymbol{x}(t) &= [\dot{\boldsymbol{x}}/|\dot{\boldsymbol{x}}| \mid \boldsymbol{x} \times \dot{\boldsymbol{x}}/|\dot{\boldsymbol{x}}| \mid \boldsymbol{x}]^{\mathsf{T}}\boldsymbol{x} && (44.28)\\ &= \begin{bmatrix} \dot{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{x}/|\dot{\boldsymbol{x}}| \\ (\boldsymbol{x} \times \dot{\boldsymbol{x}}/|\dot{\boldsymbol{x}}|)^{\mathsf{T}} \\ \boldsymbol{x}^{\mathsf{T}}\boldsymbol{x} \end{bmatrix}, && (44.29) \end{aligned}$$

where all vectors in the 3×3-matrix are unit vectors.

The first derivative in that frame is

$${}^t\dot{\boldsymbol{x}} = [\mathsf{N}(\dot{\boldsymbol{x}}),\, \boldsymbol{x} \times \mathsf{N}(\dot{\boldsymbol{x}})|,\, \boldsymbol{x}]^{\mathsf{T}}\dot{\boldsymbol{x}} = \begin{bmatrix} |\dot{\boldsymbol{x}}| \\ 0 \\ 0 \end{bmatrix}. \tag{44.30}$$

confirming (44.21). The second derivatives in that frame are

$${}^t\ddot{\boldsymbol{x}} = [\mathsf{N}(\dot{\boldsymbol{x}}),\, \boldsymbol{x} \times \mathsf{N}(\dot{\boldsymbol{x}}),\, \boldsymbol{x}]^{\mathsf{T}}\ddot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}^{\mathsf{T}}\ddot{\boldsymbol{x}}/|\dot{\boldsymbol{x}}| \\ |\boldsymbol{x}\,\dot{\boldsymbol{x}}\,\ddot{\boldsymbol{x}}|/|\dot{\boldsymbol{x}}| \\ \boldsymbol{x}^{\mathsf{T}}\ddot{\boldsymbol{x}}/|\dot{\boldsymbol{x}}| \end{bmatrix}. \tag{44.31}$$

If we only use the first two coordinates we obtain the curvature in the tangent plane

$$\begin{aligned} \kappa'(t) &= \frac{{}^t\dot{x}_1\,{}^t\ddot{x}_2 - {}^t\ddot{x}_1\,{}^t\dot{x}_2}{({}^t\dot{x}_1^2 + {}^t\dot{x}_2{}^2)^{3/2}} && (44.32)\\ &= \frac{|[\boldsymbol{x}\,\dot{\boldsymbol{x}}\,\ddot{\boldsymbol{x}}]|}{|\dot{\boldsymbol{x}}|^3} && (44.33)\\ &\le \kappa(t) && (44.34) \end{aligned}$$

thus

$$\boxed{\kappa'(t) = \frac{\sin(b\,t)\,\left(2b^2 + \cos^2(bt)\right)}{\left(b^2 + \cos^2(bt)\right)^{3/2}}}\,. \tag{44.35}$$

Hence, the curvature approximately decreases with the cosine of the distance $\pi/2 - bt$ to the north pole.

44.6 Discrete integration in the plane

We discuss two planarizations: (1) planarization of the spiral, i.e., of the curve on the sphere, and (2) planarization of the area between the branches of the spiral, each covering an azimuth of 360°.

44.6.1 Planarization of the spiral

Let the path local velocity $v(t) = \mathrm{d}s(t)/\mathrm{d}t$ and the curvature $\kappa'(t)$ be given, the task is to find the planar curve $\boldsymbol{y}(t) = (x(t), y(t))$ having the same velocity and curvature as the 3D Archimedean spiral. We start the path $\boldsymbol{x}_i$ with two points and then augment the path stepwise by one point. We fix a step size Δt, e.g., to guarantee to arrive at the north pole after N steps, then

$$\Delta t = \frac{\pi}{2bN}\,. \tag{44.36}$$

44.6.1.1 The first two points

The first point is assumed to be at the origin

$$\boldsymbol{y}_0 = \boldsymbol{0}\,. \tag{44.37}$$

The second point has a distance

$$\Delta s_t := v(t)\Delta t \tag{44.38}$$

and direction $[x_2(\Delta t), x_3(\Delta t)]$ hence, when using the approximations $\cos(t) \approx 1$ and $\sin t \approx t$:

$$\boldsymbol{y}_1 = \frac{\Delta s_0}{\sqrt{1+b^2}} \begin{bmatrix} 1 \\ b \end{bmatrix}\,. \tag{44.39}$$

44.6.1.2 Augmenting the chain

Given two points $\boldsymbol{y}_{t-1}$ and $\boldsymbol{y}_t$

$$\boldsymbol{y}_{t-1} = \begin{bmatrix} y_{t-1} \\ y_{t-1} \end{bmatrix} \quad \boldsymbol{y}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{44.40}$$

we want to determine a third point $\boldsymbol{y}_{t+1}$ which has a distance Δs_t and an offset from the prolongation depending on the curvature κ_t: The parabola, which passes through $x = -d$ and $x = 0$

$$y = \frac{1}{2}ax(x+d) \tag{44.41}$$

has curvature $\kappa = a$ at $x = -d$. Hence, the point (d, ad^2) lies on the parabola and has distance $d\sqrt{1+(ad)^2}$ to the origin, which for small products $ad \ll 1$ is a good approximation for d.

In the tangential coordinate at the tangential system t the three points therefore have the coordinates

$$
\begin{aligned}
{}^t\boldsymbol{x}_{t-1} &= |\Delta s_{t-1}| \begin{bmatrix} 1 \\ 0 \end{bmatrix} && (44.42)\\
{}^t\boldsymbol{x}_{t} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} && (44.43)\\
{}^t\boldsymbol{x}_{t+1} &= |\Delta s_t| \, \mathsf{N}\left(\begin{bmatrix} +\Delta s_t \\ \kappa'_t \, \Delta s_t^2 \end{bmatrix} \right) , && (44.44)
\end{aligned}
$$

which guarantees the curvature is close to κ'_t if the length elements $|\Delta s_{t-1}|$ and $|\Delta s_t|$ differ not too much. The 2D motion results from

$$
\boldsymbol{y} = R \, {}^t\boldsymbol{y} + \boldsymbol{t} \quad \text{with} \quad R = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \tag{44.45}
$$

with

$$
a = \frac{x_t - x_{t-1}}{\Delta s_{t-1}} \quad b = \frac{y_t - y_{t-1}}{\Delta s_{t-1}} , \qquad \boldsymbol{t} = \boldsymbol{y}_t \tag{44.46}
$$

Hence we obtain

$$
\boldsymbol{y}_{t+1} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \frac{1}{\Delta s_{t-1}} \begin{bmatrix} x_t - x_{t-1} & -(y_t - y_{t-1}) \\ y_t - y_{t-1} & x_t - x_{t-1} \end{bmatrix} \Delta s_t \, \mathsf{N}\left(\begin{bmatrix} +\Delta s_t \\ \kappa'_t \, \Delta s_t^2 \end{bmatrix} \right) . \tag{44.47}
$$

or finally

$$
\boxed{\boldsymbol{y}_{t+1} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \frac{\Delta s_t}{\Delta s_{t-1}} \begin{bmatrix} x_t - x_{t-1} & -(y_t - y_{t-1}) \\ y_t - y_{t-1} & x_t - x_{t-1} \end{bmatrix} \mathsf{N}\left(\begin{bmatrix} +\Delta s_t \\ \frac{1}{2}\kappa'_t \, \Delta s_t^2 \end{bmatrix} \right) .} \tag{44.48}
$$

with

$$
\begin{aligned}
\Delta s(t) &= \sqrt{b^2 + \cos^2(bt)} \, \Delta t && (44.49)\\
\kappa'(t) &= \frac{\sin(b\,t) \, \left(2b^2 + \cos^2(bt)\right)}{\left(b^2 + \cos^2(bt)\right)^{3/2}} && (44.50)
\end{aligned}
$$

44.6.2 Planarization of the surface region by quadrangles

The four points

$$
\begin{aligned}
[h]Q(t) &= [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{x}_3(t), \boldsymbol{x}_4(t)] && (44.51)\\
&:= [\boldsymbol{x}(t), \boldsymbol{x}(t + \Delta t), \boldsymbol{x}(t + \Delta t + 2\pi), \boldsymbol{x}(t + 2\pi)] \\
&\qquad \text{with} \quad t \le 2\pi \left(\frac{1}{4b} - 1 \right) - \Delta t
\end{aligned}
$$

build a quadrangle, see Fig. 44.4 with the having the same azimuths t and $t + \Delta t$, lying in neighbouring branches of the spiral. Therefore, they are nearly coplanar. Hence, we may map these quadrangles to the plane and concatenate them. This holds as long as the latitude bt of the upper branch is not larger than $\pi/2$, thus

$$
b(t + \Delta t + 2\pi) \le \frac{\pi}{2} \quad \text{or} \quad t \le \frac{\pi}{2b} - 2\pi = 2\pi \left(\frac{1}{4b} - 1 \right) - \Delta t \tag{44.52}
$$

For larger values t we only use triangles

$$
\mathcal{T}(t) = \left[\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{x}_3(t) \right] := \left[\boldsymbol{x}(t), \boldsymbol{x}(t + \Delta t), \boldsymbol{x}\left(\frac{\pi}{2b} \right) \right] , \tag{44.53}
$$

which connect neighbouring points on the spiral with the north pole.

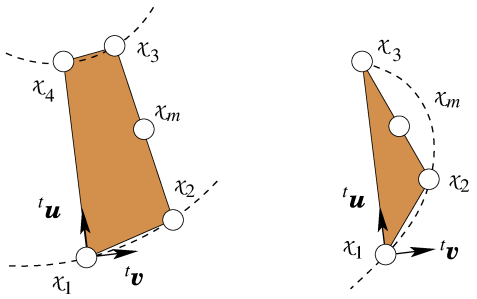


Figure 44.4: Quadrangle between two branches of the spiral and triangle close to the north pole $\boldsymbol{x}_3$

44.6.2.1 The first two points

We assume the spiral starts at the equator at $t = 0$.

We assume each quadrangle is surrounded by the four points $\boldsymbol{x}_k(t), k = 1, 2, 3, 4$ in mathematically positive orientation.

The first two points are connecting the start and the end of the first branch, hence $\boldsymbol{x}(t), t = \{0, 2\pi\}$. The first point of the first quadrangle is the image of $\boldsymbol{x}_1(0) := \boldsymbol{x}(0) = [1, 0, 0]^\mathsf{T}$ and assumed to be

$$\boldsymbol{u}_1(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{44.54}$$

The last point of that quadrangle is assumed to lie on the y axes having a distance of $|\boldsymbol{x}(2\pi) - \boldsymbol{x}(0)|$ along the meridian:

$$\boldsymbol{u}_4(0) = \begin{bmatrix} 0 \\ |\boldsymbol{x}(2\pi) - \boldsymbol{x}(0)| \end{bmatrix} . \tag{44.55}$$

44.6.2.2 Augmenting the chain

We now assume we have given the coordinates $\{\boldsymbol{u}_1(t), \boldsymbol{u}_4(t)\}$ and want to derive the coordinates $\{\boldsymbol{u}_2(t), \boldsymbol{u}_3(t)\}$.

For this, we move the quadrangle $Q(t)$ into the tangent plane, such that the first two points have xy-coordinates $\{\boldsymbol{u}_1(t), \boldsymbol{u}_4(t)\}$ in the tangent system S_t and such that the midpoint of the other two points $\boldsymbol{x}_m(t) = (\boldsymbol{x}_3(t) + \boldsymbol{x}_4(t))/2)$ lies in that plane. The third coordinates of the moved points $\{{}^t\boldsymbol{x}_3(t), {}^t\boldsymbol{x}_4(t)\}$ will be small, since the torsion of the spiral is small, and set to 0, leading to the xy-coordinates $\{\boldsymbol{u}_2(t), \boldsymbol{u}_3(t)\}$.

If we attach a (uv)-system in the plane of the triangle, with the y-axis unit vector ${}^t\boldsymbol{v}$ parallel to $\boldsymbol{x}_{14} := \boldsymbol{x}_4 - \boldsymbol{x}_1$, and the x-axis unit vector ${}^t\boldsymbol{u}$ being orthogonal to ${}^t\boldsymbol{v}$, such that the local ${}^t x_m$ coordinate is positive, then the local coordinates of the point $\boldsymbol{\chi}_k, k = 2, 3$ can be expressed as

$${}^t\boldsymbol{u}_k = \begin{bmatrix} {}^t\boldsymbol{u}^\mathsf{T} \\ {}^t\boldsymbol{v}^\mathsf{T} \end{bmatrix} \boldsymbol{x}_{1k} \quad \text{with} \quad {}^t\boldsymbol{v} = \mathsf{N}(\boldsymbol{x}_{14}) \quad \text{and} \quad {}^t\boldsymbol{u} = \mathsf{N}(\boldsymbol{v} \times (\boldsymbol{x}_m \times \boldsymbol{v})) \quad \text{and} \tag{44.56}$$

Then the points in the plane have the coordinates

$$\boldsymbol{u}_k = \boldsymbol{u}_1 + [\mathsf{N}({}^t\boldsymbol{x}_{14}^\perp) \mid \mathsf{N}({}^t\boldsymbol{x}_{14})]\, {}^t\boldsymbol{u}_k \quad \text{with} \quad \begin{bmatrix} x \\ y \end{bmatrix}^\perp = \begin{bmatrix} y \\ -x \end{bmatrix} \quad \text{and} \quad k = 2, 3\,. \tag{44.57}$$

The aggregation then uses the 2D points $(\boldsymbol{u}_2(t), \boldsymbol{u}_3(t))$ as starting point for the new quadrangle

$$\boldsymbol{u}_1(t + \Delta t) := \boldsymbol{u}_2(t) \quad \text{and} \quad \boldsymbol{u}_4(t + \Delta t) := \boldsymbol{u}_3(t)\,. \tag{44.58}$$

For triangles, we use the same procedure, setting $\boldsymbol{x}_4 := \boldsymbol{x}_3$.

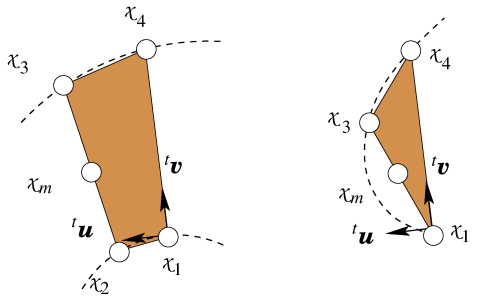


Figure 44.5: Quadrangle between two branches of the negative branch of the spiral and triangle close to the south pole $\boldsymbol{x}_1$

The same procedure can be used for the lower left (negative) part of the spiral, see Fig. 44.5 The quadrangels then are

$$\begin{aligned}[h]Q_-(t) &= [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{x}_3(t), \boldsymbol{x}_4(t)] \\ &= [\boldsymbol{x}(-t-\Delta t), \boldsymbol{x}(-t), \boldsymbol{x}(-t+2\pi), \boldsymbol{x}(-t-\Delta t+2\pi)] \\ &\quad \text{with} \quad t \geq -\frac{\pi}{2b} - \Delta t\,.\end{aligned} \tag{44.59}$$

For smaller t we again use triangles joining the south pole with two points on the spiral:

$$\mathcal{T}_-(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_3(t), \boldsymbol{x}_4(t)] := \left[\boldsymbol{x}\left(\frac{-\pi}{2b}\right), \boldsymbol{x}(-t+2\pi), \boldsymbol{x}(-t-\Delta t+2\pi)\right]\,, \tag{44.60}$$

Thus again we use the procedure for quadrangles, now, setting $\boldsymbol{x}_2 := \boldsymbol{x}_1$.

44.7 Examples

We close with some examples.

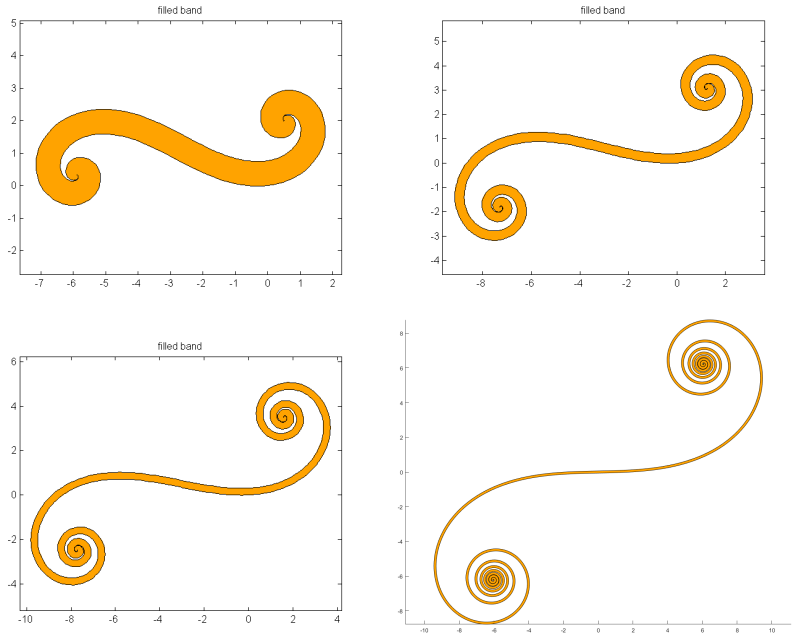


Figure 44.6: Flattened spherical archimedian spirals with $b = 2$, $b = 4$, $b = 5$ and $b = 12$ turns

Bibliography

The numbers at the end of each reference are the pages where it is cited.

Baarda, W. (1968). *A Testing Procedure for Use in Geodetic Networks*, Volume 2/5 of *Publication on Geodesy, New Series*. Delft: Netherlands Geodetic Commission. 212

Baarda, W. (1973). *S-Transformations and Criterion Matrices*, Volume 5/1 of *Publication on Geodesy, New Series*. Netherlands Geodetic Commission. 207, 208

Barath, D., M. Polic, W. Förstner, T. Sattler, T. Pajdla, and Z. Kukelova (2020a). Making Affine Correspondences Work in Camera Geometry Computation Supplementary Material. In *Proc. of ECCV*. 368

Barath, D., M. Polic, W. Förstner, T. Sattler, T. Pajdla, and Z. Kukelova (2020b). Making Affine Correspondences Work in Camera Geometry Computation. In *Proc. of ECCV*. 303, 368

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer. 7, 61, 115

Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press. 142, 145

Boyer, K. and A. Kak (1988, March). Structural Stereo for 3-D Vision. *IEEE T-PAMI 10*(2), 144–166. 15

Browne, J. (2009). Grassmann Algebra. `https://de.scribd.com/document/136872019/Grass-Mann-Algebra-Book`, last visited 12.4.2024. 432, 433

Brügelmann, R. and W. Förstner (1992). Noise Estimation for Color Edge Extraction. In W. Förstner and S. Ruwiedel (Eds.), *Robust Computer Vision*, pp. 90–107. Wichmann, Karlsruhe. 383

Castleman, K. R. (1996). *Digital Image Processing*. Prentice Hall Inc. 35, 41, 44

Cayley, A. (1846). Sur quelques proprétés des déterminants gauches. *Journal für die reine und angewandte Mathematik 32*, 119–123. 476, 477

Cooley, J. W. and J. W. Tukey (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation 19*, 297–301. 39

Cover, T. and J. A. Thomas (1991). *Elements of Information Theory*. USA: John Wiley & Sons. 7

Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM. 52

Eade, E. (2014). Lie Groups for 2D and 3D Transformations. `http://ethaneade.com/lie_groups.pdf`, last visited 2.6.2016. 234

Eichhardt, I. and D. Chetverikov (2018). Affine Correspondences Between Central Cameras for Rapid Relative Pose Estimation. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Eds.), *Computer Vision – ECCV 2018*, Cham, pp. 488–503. Springer International Publishing. 393, 394, 413

Faugeras, O. and Q.-T. Luong (2001). *The Geometry of Multiple Images*. MIT Press. with contributions from T. Papadopoulo. 432, 439

Faugeras, O. and T. Papadopoulo (1998). Grassmann-Cayley Algebra for Modeling Systems of Cameras and the Algebraic Equations of the Manifold of Trifocal Tensors. In *Trans. of the Royal Society A, 365*, pp. 1123–1152. 432

Folland, G. B. and A. Sitaram (1997). The Uncertainty Principle: A mathematical Survey. *The Journal of Fourier Analysis and Applications 3*(3), 207–238. 57

Förstner, W. (1982). Systematic Errors in Photogrammetric Point Determination. In *Proc. Survey Control Networks, International Federationof Surveyors (FIG), Meeting Study Group 5B*, Denmark, pp. 197–209. 166

Förstner, W. (1985). Determination of the Additive Noise Variance in Observed Autoregressive Processes Using Variance Component Estimation Technique. *Statistics and Decisions Supplement Issue 2*, 263–274. 21

Förstner, W. (1991). *Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung*. 382

Förstner, W. (1993). *Image Matching*, Volume II, Chapter 16, pp. 289–379. Addison Wesley. 329

Förstner, W. (1998). On the Theoretical Accuracy of Multi Image Matching, Restoration and Triangulation. In *Festschrift Univ. Prof. Dr.-Ing.-Dr, h. c. mult. Gottfried Konecny zur Emeritierung*, pp. 105–118. Wiss. Arbeiten der Fachr. Vermessungswesen der Univ. Hannover. 333, 468, 469

Förstner, W. (1999). Uncertain Neighborhood Relations of Point Sets and Fuzzy Delaunay Triangulation. In W. Förstner, J. M. Buhmann, A. Faber, and P. Faber (Eds.), *Mustererkennung 1999*, Informatik aktuell, pp. 213–222. 21. DAGM–Symposium, Bonn: Springer-Verlag. 293

Förstner, W. (2000). Image Preprocessing for Feature Extraction in Digital Intensity, Color and Range Images. In *Geomatic Methods for the Analysis of Data in Earth Sciences*, Volume 95/2000 of *Lecture Notes in Earth Sciences*, pp. 165–189. Springer. 364, 372, 381, 383

Förstner, W. (2010). Minimal Representations for Uncertainty and Estimation in Projective Spaces. In *Proc. of Asian Conference on Computer Vision, Queenstown*. 207, 208

Förstner, W. (2013, 08). Graphical Models in Geodesy and Photogrammetry. *Zeitschrift für Photogrammetrie, Fernerkundung, Geoinformation 2013*(4), 255–267. 219

Förstner, W., A. Brunn, and S. Heuel (2000). Statistically Testing Uncertain Geometric Relations. In G. Sommer, N. Krüger, and C. Perwass (Eds.), *Mustererkennung 2000*, Informatik aktuell, pp. 17–26. 22. DAGM Symposium, Kiel: Springer. 432

Förstner, W., T. Dickscheid, and F. Schindler (2009). Detecting Interpretable and Accurate Scale-Invariant Keypoints. In *12th IEEE International Conference on Computer Vision (ICCV'09)*, Kyoto, Japan, pp. 2256–2263. 159

Förstner, W. and K. Khoshelham (2017). Efficient and Accurate Registration of Point Clouds with Plane to Plane Correspondences. In *3rd International Workshop on Recovering 6D Object Pose*. 234

Förstner, W. and B. P. Wrobel (2016). *Photogrammetric Computer Vision – Statistics, Geometry, Orientation and Reconstruction*. Springer. 7, 32, 106, 120, 144, 158, 159, 189, 207, 208, 209, 210, 217, 233, 247, 253, 272, 274, 371, 374, 384, 413, 428, 432, 433, 452, 458, 460, 485

Georgeff, M. P. and C. S. Wallace (1984). A General Selection Criterion for Inductive Inference. In *Advances in Artificial Intelligence*, pp. 473–482. Proc. European Conf on Artificial Intelligence, North Holland Amsterdam. 7

Gonzales, R. C. and P. Wintz (1977). *Digital Image Processing*. Advanced Book Programm. Addison-Wesley. 34

Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme. (Erste Mitteilung). *Mathematische Annalen 69*, 331–371. 52

Haddon, J. and D. A. Forsyth (2001, July). Noise in Bilinear Problems. In *ICCV'01, Vol. II*, pp. 622–627. Proc. of 8th Intl. Conf. on Computer Vision, Vancouver, BC: IEEE Computer Society. 192

Hartley, R. I. and A. Zisserman (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press. 201

Hestenes, D. and R. Ziegler (1991). Projective Geometry with Clifford Algebra. *Acta Applicandae Mathematicae 23*, 25–63. 433

Hjalmars, S. (1977). Evidence for Boltzmann's H as a capital eta. *American Journal of Physics 45*(2), 214–215. 11

Jähne, B. (1999). Local Structure. In B. Jähne, H. Horst Haussecker, and P. Peter Geissler (Eds.), *Handbook of Computer Vision and Applications*, Chapter 10, pp. 209–238. Academic Press. 370

Jähne, B., H. Scharr, and S. Körkel (1999). Principles of Filter Design. In B. Jähne, H. Horst Haussecker, and P. Peter Geissler (Eds.), *Handbook of Computer Vision and Applications*, Chapter 6, pp. 125–151. Academic Press. 35

Jordan, M. I. (1995). Why the logistic function? A tutorial discussion on probabilities and neural networks. `https://cs.nyu.edu/~roweis/csc2515-2006/readings/whylogistic.pdf`, last visited 12.4.2024. 61

Katkar, L. N. (2019). extensions of the theorem of pythagoras. In *Azim Premji University At Right Angles*. 474

Koch, K.-R. (1999). *Parameter Estimation and Hypothesis Testing in Linear Models* (2nd ed.). Springer. 166, 194, 195, 385

Koecher, M. and R. Remmert (1991). Chapter 7, Hamilton's Quaternions; Chapter 9, CAYLEY Numbers or Alternative Division Algebras; Chapter 10, Section 1, Composition Algebras. In H. H. F. Hirzebruch M. Koecher K. Mainzer J. Neukirch A. Prestel H.-D. Ebbinghaus and R. Remmert (Eds.), *Numbers*. Springer-Verlag. 474

Koenderink, J. J. and A. J. v. Doorn (2002). Image Processing Done Right. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, London, UK, UK, pp. 158–172. Springer-Verlag. 322

Kolbe, T. H. (2000). *Identifikation und Rekonstruktion von Gebäuden in Luftbildern mittels unscharfer Constraints*. Shaker Verlag: Shaker. 7, 19

Kullback, S. and R. A. Leibler (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics 22*(1), 79 – 86. 13

Läbe, T. and W. Förstner (2006). Automatic Relative Orientation of Images. In *Proceedings of the 5th Turkish-German Joint Geodetic Days*. 410

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 2278–2324. 84, 90

Leonardos, S., C. Allen-Blanchette, and J. Gallier (2015). The exponential map for the group of similarity transformations and applications to motion interpolation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 377–382. 241

Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics 2*(2), 164–168. 116

Mangelson, J. G., M. Ghaffari, R. Vasudevan, and R. M. Eustice (2019). Characterizing the Uncertainty of Jointly Distributed Poses in the Lie Algebra. *IEEE Transactions on Robotics 36*, 1371–1388. 234

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics 11*(2), 431–441. 116

McGillem, C. D. and M. Svedlow (1976). Image Registration Error Variance as a Measure of Overlay. *IEEE Geoscience Electronics 14*, 44–49. 339

Molenaar, M. (1981). *A further inquiry into the theory of S-transformations and criterion matrices*. New Series 26. Netherlands Geodetic Commission NCG: Publications on Geodesy, Delft, Rijkscommissie voor Geodesie. 208

Nielsen, M. (2017). *Neural Networks and Deep Learning*. `http://neuralnetworksanddeeplearning.com/`, last visited 12.4.2024. 75, 76, 78, 80, 81

Papoulis, A. (1965). *Probability, Random Variables and Stochastic Processes* (4th ed.). McGraw-Hill. 43

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes* (2nd ed.). Electrical Engineering. McGraw-Hill. 7, 32

Pepić, S. H. (2010). Weighted Moore-Penrose Inverse: PHP vs. Mathematica. *Ser. Math. Infrom. 25*, 35–45. 149

Perwass, C. B. U. (2000). *Applications of Geometric Algebra in Computer Vision*. Ph. D. thesis, Cambridge University. 433

Rader, C. (1968). Discrete Fourier transforms when the number of data samples is prime. *Proceedings of the IEEE 56*(6), 1107–1108. 39

Rao, R. C. (1967). Least squares theory using an estimated dispersion matrix and its application to measurement of signals. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*, pp. 355–372. Univ. of Calif. Press. Lemma 5a. 162, 163, 164, 182

Roos, T., H. Wettig, P. D. Grünwald, P. Myllymäki, and H. Tirri (2005). On Discriminative Bayesian Network Classifiers and Logistic Regression. *Machine Learning 59*, 267–296. 61

Savage, L. J. (1972). *The Foundations of Statistics*. Dover Publications. 9

Schneider, J. and W. Förstner (2013). Bundle Adjustment and System Calibration with Points at Infinity for Omnidirectional Camera Systems. *Zeitschrift für Photogrammetrie, Fernerkundung und Geoinformation 4*,

309–321. 250

Shannon, C. E. and W. Weaver (1949). *The Mathematical Theory of Communication*. Urbana, Illinois: The University of Illinois Press. 8, 11, 23, 49, 50

Shu, X. (2013). Bicubic Interpolation. Dept. Electrical and Computer Engineering, McMaster, Canada, https://www.ece.mcmaster.ca/~xwu/3sk3/interpolation.pdf. 376

Solà, J., J. Deray, and D. Atchuthan (2018). A micro Lie theory for state estimation in robotics. *CoRR abs/1812.01537*. 232

Steeb, W.-H., Y. Hardy, and R. Stoop (2003, jun). Discrete wavelets and perturbation theory. *Journal of Physics A: Mathematical and General 36*(24), 6807. 54

Strecha, C., W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen (2008). On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*. 410

Takahashi, K., J. Fagan, and M.-S. Chen (1973). Formation of a sparse bus impedance matrix and its application to short circuit study. In *IEEE Power Engineering Society*, Volume 7. 169

Vanhatalo, J. and A. Vehtari (2008). Modelling local and global phenomena with sparse Gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*, pp. 571–578. 169

Viola, P. and M. Jones (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Volume 1, pp. I–I. 56

Vosselman, G. (1992). *Relational Matching*, Volume 628 of *Lecture Notes on Computer Science*. Springer Verlag. 7, 15

Waegli, B. (1998). Investigations into the Noise Characteristics of Digitized Aerial Images. In *Intl. Archives of Photogrammetry and Remote Sensing*, Volume 32–2, pp. 341–348. Proc. of ISPRS Comm. II Symposium, Cambridge, UK. 384

Weisstein, E. W. (2023). Haar Function. From MathWorld–A Wolfram Web Resource. 53

Weisstein, E. W. (2022). Calculus of Variations. From MathWorld–A Wolfram Web Resource. 16

Whittaker, J. M. (1915). On the functions which are represented by the expansions of the interpolation-theory. *Proc. Roy. Soc. 35*, 181–194. 49

Winograd, S. (1976). On computing the Discrete Fourier Transform. *Proceedings of the National Academy of Sciences of the United States of America 73*(4), 1005–1006. 39