10 Pros and Cons Against Performance Characterization of Vision Algorithms

Wolfgang Förstner
Institut für Photogrammetrie, Universität Bonn
Nußallee 15, D-53115 Bonn, e-mail: wf@ipb.uni-bonn.de
Workshop 'Performance Characteristics of Vision Algorithms', Cambridge, 1996

Abstract

The paper discusses objections against performance characterization of vision algorithms and explains their motivation. Short and long-term arguments are given which overcome these objections. The methodology for performance characterization is sketched to demonstrate the feasibility of empirical testing of vision algorithms.

Contents

0 Motivation **2**

1 Evaluation is task dependent **3**
1.1 Pro . . . 3
1.2 Contra: Characterize performance and select adequate algorithm . . . 3

2 Vision is only one module **4**
2.1 Pro . . . 4
2.2 Contra: Design *Traffic Light Programs* . . . 5
2.3 Example: Edge extraction . . . 5
2.3.1 Precision . . . 5
2.3.2 Accuracy . . . 6
2.3.3 Reliability . . . 6

3 Vision is too complex **7**
3.1 Pro . . . 7
3.2 Contra: Modularize . . . 7
3.3 Combining Probabilities and Weights . . . 8
3.3.1 Combining Probabilities . . . 8
3.3.2 Combining Estimates . . . 8

4 The used models are wrong **9**
4.1 Pro . . . 9
4.2 Contra: Usefulness of models is decisive . . . 9
4.3 Examples . . . 9
4.3.1 Theoretical analysis for neglecting parameters . . . 9
4.3.2 Non-Gaussian distributions due to modeling errors . . . 10
4.3.3 Correlations and tolerances . . . 10
4.3.4 Modeling background . . . 12

5 Measures are not comparable **12**
5.1 Pro . . . 12
5.2 Contra: Use statistical measures . . . 12
5.3 Examples . . . 12
5.3.1 Statistical interpretation of regularization . . . 12
5.3.2 Dependency of covariance matrices on the coordinate system . . . 13

6 No theory for algorithms **15**
6.1 Pro . . . 15
6.2 Contra: Performance prediction stimulates theoretical research . . . 15

7 Too many tuning parameters **16**
7.1 Pro . . . 16
7.2 Contra: Only accept meaningful tuning parameters . . . 16

8 Ground truth is too expensive **17**
8.1 Pro . . . 17
8.2 Contra: Share costs . . . 17
8.3 Example: Costs for testing orientation software . . . 17

9 Simulations are not reality **18**
9.1 Pro . . . 18
9.2 Contra: Simulations replace complicated theory . . . 18
9.3 Example: Evaluating segmentation results . . . 18

10 Testing is not acknowledged **20**
10.1 Pro . . . 20
10.2 Contra: Empirical testing is worthwhile . . . 21

11 Conclusions **21**

0 Motivation

For at least 10 years Computer Vision has been confronted with papers and discussions on the scientific value of its results and the difficulties in transferring the results to practical systems.

A change of awareness seems to have happened: An the Computer Vision Workshop 1985 two controversial papers, with different view, agreed on the *lack of theoretical research* ([Haralick 1985], [Price 1985]), which should go along with the development of vision procedures: experimental proofs are not enough.

The dialogue on 'Ignorance, Myopia, and Naivité in Computer Vision Systems' initiated by R. Jain and T. Binford ([Jain 1991]) and the responses documented the necessity of evaluating theoretical findings, vision procedures algorithms etc. by *using empirical data* in order to increase the number of real world applications of Computer Vision Research.

When observing the increasing number of papers which propose new solutions to classical problems, especially using increasingly more demanding theoretical tools, it seems to become clear that empirical testing of vision algorithms is necessary to allow a clear comparison of the proposed methods by the users of such algorithms. Together with the underlying theories a clear performance characterization of algorithms is necessary.

When discussing the necessity of empirical testing and performance characterization a number of strong objections are posed repeatedly. Their honesty cannot be debated. This requires a serious attempt to find out their truth, but also to show either their shortsightedness or the means to overcome these objections.

The purpose of this paper is to collect the most commonly posed objections against performance characterization. Each of them is correct to some extent. The range of their validity is discussed and opposed with a view to allow the start of accepting and applying formal quality assessment. In each case examples are given to demonstrate that the posed objections can be overcome. Though no commonly accepted methodology seems to be available and is not meant to be proposed here, the discussion should provide a strong motivation for developing vision algorithms with clearly defined performance characterization based on both, theoretical research and empirical testing.

1 Evaluation is task dependent

1.1 Pro

The evaluation of vision algorithms is task dependent. Vision modules always are part of an application.

There is no such thing as a vision algorithm per se. E. g. edge detection never is a goal on its own. Vision algorithms are designed to solve a task. The variety of tasks makes it necessary to choose the best algorithms or adapt existing algorithms in order to fulfill the constraints of the application e. g. with respect to resolution, time or space requirements.

The variety of tasks leads to a variety of requirements. Therefore no single set of constraints can be specified allowing to give a limited set of basic algorithms. E. g. edge detection may aim at precision, accuracy, resolution, noise insensitivity, reliability, speed, etc. Criteria on the measures for all of these quality notions in general are different for different applications, making a recommendation of a certain algorithm obsolete.

1.2 Contra: Characterize performance and select adequate algorithm

The same vision module may be part of several applications. This may be just to distribute the cost for its development or to be able to reuse the software, possibly much later. Therefore an inversion is necessary: The developer is responsible for the specification of a set of useful quality *measures*, which are variables, not values. The dependency of these quality measures on the characteristics of the image data needs to be investigated and reported in order to enable the user of the algorithms to decide on the usefulness of the algorithm for the specific application, which might not have been foreseen by the developer of the algorithm.

Formally the result r of an algorithm a depends on the input data d and the tuning parameters t, thus $r = r(d; a, t)$. The specification consists of requiring r to be achieved with a quality $q(r)$ better than some value q_0, or

$$q(r|d; a, t) \geq q_0 \tag{1}$$

assuming q to increase with increasing quality. If q is vector valued the requirement in (1) refers to each individual component.

Usually a subset $\mathcal{D}_s = \{d_1, d_2, ...\} \subset \mathcal{D}$ of representative input data is given, which can be seen to be a set of samples of a stochastic variable $\underline{d}$. If the characteristics of this set is estimated from the given subset one can theoretically derive the expected quality and change the requirement into

$$\mathsf{E}(\underline{q}(\underline{r}|\underline{d}, a, t)) \geq q_0 \tag{2}$$

or if the user allows that the requirements are fulfilled only with a minimum probability P_0

$$P(\underline{q}(\underline{r}|\underline{d}, a, t) \geq q_0) \geq P_0 \tag{3}$$

This now allows to explicitly write down admissible algorithms $\widehat{a}$ with tuning parameters $\widehat{t}$

$$\{\widehat{a},\widehat{t}\} = \{(a,t)|P(\underline{q}(\underline{r}|\underline{d},a,t) \geq q_0) \geq P_0\} \tag{4}$$

This reasoning leads to a set of very clear conclusions, discussed in the order of their appearance while choosing an admissible algorithm:

- The quality evaluation functions need to be chosen in such a way that they are theoretically and algorithmically tractable and that they are acceptable by the user. This is no severe restriction as most users would refer to standards in quality control, e. g. using standard deviations, well defined tolerances or relative frequencies.
- The requirements, thus the values q_0 and P_0 are to be specified by the user needs, thus may vary from application to application.
- The characteristics of the complete set $\mathcal{D}$ of possible input data need to be found.

  This is usually based on the given training data *together with* some rules on how to generalize the data and *only* with respect to the task in concern. At this stage *learning* is required.

  The characterization only needs to be performed up to the point which is relevant for the calculation of $P(q \geq q_0)$, which, due to the strong projection taking place, is much more likely to be feasible than the general characterization of the data. No complete specification of the input data is necessary at all, which is not possible anyway. This characterization may use any type of representation, e. g. algebra or tables, and may be derived theoretically or by simulation, e. g. bootstrapping. Of course, algebraic results are more valuable due to their generalization capability, however, function approximations in all cases are well suited if the domain of their validity is well documented.

  E. g. when restoring images, the statistics of signal and noise may be sufficiently described by the power spectra, both derivable from a small set of given images under quite general conditions. But for extracting edges a characterization of their form may be necessary, without, however, neglecting their origin (shadow, contour, illumination, etc.).
- The derivation of the distribution of q may again use analytical or simulation techniques.
- Choosing *one* algorithm to satisfy the quality requirements may at the same time take other constraints into account.

Obviously the characterization of an algorithm with respect to a set of standard quality measures and a set $\{\mathcal{D}_i\}$ of classes of input data would enable users to invert the simulations provided by the author of the algorithm and to select the algorithm fitting to the application, which the author need not have thought of ahead.

Characterizing images with respect to certain tasks therefore is a key issue in performance characterization.

2 Vision is only one module

2.1 Pro

Vision modules are usually only a small part within a system, e. g. identifying the type and position of a part on a conveyor belt within an assembly line, or determining the exterior orientation within a system for rectifying aerial photos to map scale. Evaluation of the performance of such a module needs to be interfaced with the requirements of the complete sequence of modules which makes characterization with respect to a task outside the vision module difficult. This is a strong variation of the previous objection.

2.2 Contra: Design *Traffic Light Programs*

Each module within a system, however, needs to know its own capabilities. This also holds for the vision modules.

This has several consequences:

1. The vision module needs to contain tools for *self diagnosis*. This means it needs to be able to estimate its own performance. Together with the result it should produce useful values for characterizing the quality of the result.
2. Therefore the vision module needs to *know its own limitations*. In case of failure, the module should indicate this and give possible causes for the failure. This would enable the calling routine to react properly.
3. In order for the module to be able to perform such a kind of self diagnosis, quality measures need to be part of output and input specification for a vision module.

As a consequence, vision modules should be so-called *traffic light programs*, with well defined output:

green: The result is correct. Its quality is specified.

yellow: The result may be correct. It needs to be checked, possible errors need to be specified. In case it is correct, its quality is specified.

red: No result has been achieved or it certainly is incorrect. Possible causes for the failure are specified.

2.3 Example: Edge extraction.

An example for such a characterization of performance is given for edge detection. The example assumes the only task of the vision module is to detect and measure the position of an edge, e. g. for visual inspection.

The following performance measures can be theoretically derived from the edge extraction procedure and be used to feed a traffic light program, which just compares the achieved performance with the specifications.

2.3.1 Precision

The precision of extracted edges can be characterized by their standard deviation across the edge and the standard deviation of the orientation.

The precision of an edge can easily be derived in case we treat edge location as template matching. We obtain for the variance σ_u^2 of the position across the edge (cf. [Förstner 1992])

$$\sigma_u^2 = \frac{\sigma_n^2}{\sum_{r,c} f_u^2(r,c)} \tag{5}$$

where σ_n is the standard deviation of the noise, f_u is the derivative of the template edge across the edge, i. e. in u-direction, and the sum is to be taken over the template window. Similarly we obtain the variance σ_ϕ^2 of the orientation

$$\sigma_\phi^2 = \frac{\sigma_n^2}{\sum_{r,c} v^2(r,c) \cdot f_u^2(r,c)} \tag{6}$$

where v is the coordinate of the pixel along the edge. We now assume square $n \times n$ windows to be used. The average squared gradient is defined as $\sigma_{f_u}^2 \doteq \sum_{r,c} f_u^2/(n \cdot s)$ and only relates to the $n \cdot s$ pixels along the edge of width s. Observe $\sigma_{f_u}^2$ to represent the squared gradient magnitude in case the gradient is constant along the edge.

Then we can simplify the standard deviations to

$$\sigma_u = \frac{1}{\sqrt{n \cdot s}} \frac{\sigma_n}{\sigma_{f_u}} \qquad \sigma_\phi = \frac{1}{\sqrt{n \cdot s}} \sqrt{\frac{12}{n^2 - 1}} \frac{\sigma_n}{\sigma_{f_u}} \tag{7}$$

Approximating the length $l = n$ and assuming constant noise variance σ_n^2 and width s, with the general relation between standard deviations and corresponding weights

$$w_i = \frac{\sigma_0^2}{\sigma_i^2} \tag{8}$$

this leads to the weights of the position across the edge and the orientation:

$$w_u = l \cdot \sigma_{f_u}^2 \qquad w_\phi = \frac{1}{12} l^3 \cdot \sigma_{f_u}^2 \tag{9}$$

Observe the weight of the position goes proportional to the length, whereas the weight of the orientation goes with the third power of the length of the edge, both weights go with the squared gradient magnitude, here represented by $\sigma_{f_u}^2$.

These weights may be used to derive the covariance matrix of the 4 coordinates specifying a straight line segment.

Of course a similar derivation of the precision of edges can be performed for other edge extraction schemes. The important point here is: the variances – and possibly covariances – can be used in subsequent steps of the image analysis.

2.3.2 Accuracy

The internal precision may be misleading in case one has to fear systematic errors causing a bias in the position of the edge position. Accuracy, e. g. μ_u for the edge position across the edge, then can be described by the variance which takes the bias into account:

$$\mu_u^2 = \sigma_u^2 + b_u^2 \tag{10}$$

where b_u is the expected bias across the edge.

E. g. in case the edge is circular with curvature $\kappa = 1/r$ the bias in edge position is the deviation between the true edge point on the curved edge and the mean edge position lying inside the circle approximating the curved edge. It depends on the curvature and the length l of the edge and can be approximated by:

$$b_u = \int_{-l/2}^{l/2} \frac{1}{2} \kappa x^2 dx / l = \frac{1}{24} \kappa \cdot l^2 \tag{11}$$

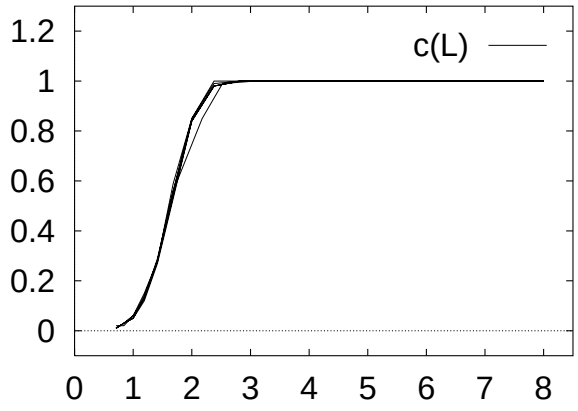
2.3.3 Reliability

The reliability of edge extraction can be measured by the probability that a pixel is classified as an edge pixel in case it actually is one. As the classification of pixels into edge and non-edge pixels is usually performed by thresholding the gradient magnitude, and this procedure can be interpreted to be a *hypothesis test* on the gradient to be significantly nonzero, the *power* of the test can be used to characterize the reliability of edge detection, while the significance level immediately gives the probability of detecting edge pixels where ther actually are no adges.

Assuming the noise to be Gaussian with mean 0 and standard deviation σ_n, the threshold on $T = g_u / \sigma_{g_u}$ to be $k(\alpha)$ depending on the significance number α of the test, and the true edge leads to a gradient magnitude being a factor δ (non-centrality parameter of the non-central Normal-distribution) larger than σ_{g_u}, the power of the test is given by (cf. [Förstner 1987]):

$$\begin{aligned} \beta(\delta, \alpha) &= P(|T| > k | \text{pixel is edge}) \\ &= 1 - \Phi(k(\alpha) - \delta) + \Phi(k(\alpha) + \delta) \end{aligned}$$

Figure 1: The expected coverage $c(L)$ of an extracted edge in dependency on the signal to noise ratio SNR $= contrast/\sigma_n$. For SNR > 2.5 one can expect the line to be extracted without gaps. The figure shows the result of 6 different experiments. The significance level is 95 %. (from FUCHS et al. 1994).



with the normalized Gaussian distribution $\Phi(x)$. The standard deviation σ_{g_u} of the gradient magnitude across the edge depends on the noise level σ_n and the function $g_u = f(g)$, e. g. the convolution kernel, to determine the gradient.

The power function β can be interpreted as the *coverage* $c(\mathsf{L})$ of a long edge. It specifies the number of edge pixels of a long edge actually being detected with respect to the length of the line in pixels and is given by

$$c(\mathsf{L}) = \frac{\#\text{edge pixels found}}{\text{length of edge[pel]}} = \beta(\delta, \alpha) \tag{12}$$

An example for an *empirically* derived line coverage is given in Fig. 1. It is taken from [Fuchs *et al.* 1994], where a complete analysis of the quality of a polymorphic feature extraction scheme (cf. [Förstner 1994b]) is given.

3 Vision is too complex

3.1 Pro

Vision systems are not monolithic. They usually consist of many, partly small, algorithms. The interaction between these algorithms is usually data dependent. The evaluation of a complex network of algorithms seems to be intractable. Even in case the aforementioned quality measures are available, propagation of quality measures through a network is not feasible.

3.2 Contra: Modularize

Modularization is a classical scheme in systems design. Modularization is also necessary in order to make performance predictable.

As arbitrarily complex networks are not tractable it is useful to define levels of abstraction in vision systems, i. e. to define an aggregation hierarchy of vision modules, which of course is task specific. Then at each aggregation level quality can be combined using the quality results of the individual modules and allow performance evaluation of the combination of the output of the individual modules. This at the same time compensates for the non-optimality of quality measures used in the individual modules. This

is comparable to the hierarchical structures in decision making processes e. g. in large agencies, where the group leader integrates the results of the individuals, based on the larger context which is available.

3.3 Combining Probabilities and Weights

3.3.1 Combining Probabilities

In classification schemes often only relative probabilities can be estimated, e. g. $p(\omega_i|d) \sim p(d|\omega_i) \cdot p(\omega_i)$ leading to a vector of likelihoods, which does not sum to 1, due to the – local – lack of knowledge about the total space of alternatives. In case the calling routine has this knowledge, normalization leads to (conditional) probabilities which sum to 1, following the basic relation of Bayes.

3.3.2 Combining Estimates

The situation is not so clear in case of parameter estimation. Here individual results, say $x_i, i = 1, ..., n$ with standard deviations σ_i need to be combined. We assume some estimation processes lead to the individual x_i, based on original measurements. The classical scheme is to combine the x_i using the weights from (8) with an *arbitrary* reference variance σ_0^2, in order to obtain e. g. the weighted mean

$$\widehat{x} = \frac{\sum_i w_i x_i}{\sum_i w_i} \tag{13}$$

With the residuals $e_i = \widehat{x} - x_i$ the estimated reference variance

$$\widehat{\sigma}_0^2 = \frac{\sum_i w_i e_i^2}{n-1} \tag{14}$$

can be testet, as

$$\frac{\widehat{\sigma}_0^2}{\sigma_0^2} \sim F_{n-1,\infty} \qquad E\left(\frac{\widehat{\sigma}_0^2}{\sigma_0^2}\right) = 1 \tag{15}$$

if and only if the given standard deviations σ_i or the weights w_i *are correct.*

There are many reasons why this test in general will not be accepted, i. e. why $\widehat{\sigma}_0^2/\sigma_0^2 >> 1$:

- gross errors or blunders in the observations leading to x_i
- a wrong model for estimating the x_i
- wrong weights used for deriving the x_i
- neglected correlations
- deviations from the assumption of normality in the distribution of the observations used for estimating the x_i

However, *independent on the cause* for the empirical reference variance to be much larger than 1, due to

$$\widehat{\sigma}_i^2 = \frac{\widehat{\sigma}_0^2}{\sigma_0^2}\sigma_i^2 \tag{16}$$

we may use the updated weights

$$w_i^{(\text{new})} = \frac{\sigma_0^2}{\widehat{\sigma}_0^2} w_i \tag{17}$$

in the following steps, as these are more realistic than the old weights from the individual steps.

Observe, that the weighted mean from (13) is independent on the chosen reference variance, thus would yield the same value if the new weights would have been chosen.

Experiences with longer chains of image analysis steps (cf. e. g. [Förstner 1994a]) confirm the possibility to link suboptimal partial results without losing the ability to evaluate the final result, in spite of the submodules containing severe nonlinearities.

4 The used models are wrong

4.1 Pro

One of the most frequent objections against quantitative performance characterization can be summarized as: the used models are wrong, so any type of formal evaluation is not valid. Examples for this kind of objection are 'the Gaussian model for noise does not hold', 'you neglect this and that effect', 'you do not calibrate your camera properly', 'background clutter cannot be captured by model', etc.

4.2 Contra: Usefulness of models is decisive

Actually all these objections are correct, as *all models are wrong* in a strict sense. However, only the usefulness not the correctness of the models with respect to a specific task is relevant. This is good engineering tradition. Moreover, models are necessary in order to be able to predict performance and to be able to properly design systems.

Therefore one must accept the sub-optimality of models. The degree of sub-optimality needs to be analyzed theoretically, e. g. by showing the bias to be much smaller than the standard deviation, or by showing the variance of certain effects to be small enough to neglect them within the model. On the other hand, in most cases no optimal solution is required but only an acceptable one, thus replacing optimization by constraint satisfaction problems.

4.3 Examples

4.3.1 Theoretical analysis for neglecting parameters

Assume depth being determined using stereo. In the most simple case we can determine depth by:

$$z = c\frac{b}{p_x} \tag{18}$$

where c is the focal length, b the length of the base line and p_x the horizontal parallax. The task is to decide whether the uncertainty of c and b can be neglected when predicting the precision of z, which would yield the well known relation

$$\sigma_z = \frac{cb}{p_x^2}\sigma_{p_x} = \frac{z^2}{cb}\sigma_{p_x} \tag{19}$$

indicating the precision decreasing with the square of the distance.

The relative precision σ_z/z of the depth in case of uncorrelated c, b and p_x is given by

$$\left(\frac{\sigma_z}{z}\right)^2 = \left(\frac{\sigma_c}{c}\right)^2 + \left(\frac{\sigma_b}{b}\right)^2 + \left(\frac{\sigma_{p_x}}{p_x}\right)^2 \tag{20}$$

depending on the relative precision of the focal length the basis and the parallax. Assume we know the geometry of the setup and have standard deviations at hand:

$$c = 20 \text{ mm} \quad \sigma_c = 0.01 \text{ mm} \tag{21}$$

$$b = 300 \text{ mm} \quad \sigma_b = 0.5 \text{ mm} \tag{22}$$

$$p_x = 4 \text{ mm} \quad \sigma_{p_x} = 0.005 \text{ mm} \tag{23}$$

we can follow:

$$\begin{aligned}\left(\frac{\sigma_z}{z}\right)^2 &= \left(\frac{0.01}{20}\right)^2 + \left(\frac{0.5}{300}\right)^2 + \left(\frac{0.005}{4}\right)^2 \\ &= \left(\frac{1}{2000}\right)^2 + \left(\frac{1}{600}\right)^2 + \left(\frac{1}{800}\right)^2\end{aligned}$$

indicating that under these conditions, the inaccuracy of c can be neglected when predicting the precision of the depth, however, the inaccuracy of b needs to be taken into account.

4.3.2 Non-Gaussian distributions due to modeling errors

Image noise usually is modeled to be Gaussian, causing problems in argumentation as intensities are non-negative and discrete. Even in case one accepts the Gaussian to be a continuous approximation to a discrete distribution the empirical histogram of gradients of homogeneous regions of varying intensity – excluding pixels in edge regions – shows clear deviations from the Gaussian density: it is long tailed.

However, this is not an indication that the Gaussian assumption cannot be used for modeling noise in principle. The following reasoning shows that the signal dependency of the noise variance is the cause for the long tailed behavior of the distribution of the gradients.

The noise variance in a digital image can in a first approximation be modeled as $\sigma_n^2 = a+bg$, where g is the mean intensity of a pixel and a and $b \geq 0$ are some coefficients, a roughly representing electronic noise and rounding errors and bg representing the Poisson statistics of the photon flux.

Now assume two regions of size A_1 and A_2 with different intensity. If $b \neq 0$ their noise variance will be different leading to Gaussian distribution $N(0, \sigma_1^2)$ and $N(0, \sigma_2^2)$. Thus their joint distribution is a mixture of two Gaussians with density

$$f(x) = a_1\phi(x|0, \sigma_1^2) + a_2\phi(x|0, \sigma_2^2) \tag{24}$$

using $a_i = A_i/(A_1 + A_2)$. This is always long tailed with curtuosis

$$\begin{aligned}\eta &= \frac{E(x^4)}{3\ \sigma^4} \\ &= 1 + a_1a_2\frac{(\sigma_1 - \sigma_2)^2(\sigma_1 + \sigma_2)^2}{(a_1\sigma_1^2 + a_2\sigma_2^2)^2} \geq 1\end{aligned}$$

This shows that, independent on the reason, distributions always tend to be long-tailed. On the other hand, longtailedness of a distribution may give rise to the question whether the error model should be refined by assuming a mixture density.

4.3.3 Correlations and tolerances

The internally predicted accuracies in nearly all cases are too optimistic. This especially holds for the predicted variances from estimation processes. This can be theoretically motivated, as the inverted normal equation matrix representing the covariance matrix of the estimates is the Cramer-Rao-bound on the efficiency of the estimate, thus stating that the result will not be better than the predicted covariance matrix.

However, in many cases only the variances of the result are given and used in the following steps. This corresponds to using the variances and neglecting the nearly always occuring correlations.

A simple example demonstrates the severe effect of neglecting correlations.

Assume range data to be influenced by two effects: miscalibration and random noise. Assume both effects to be of random nature, thus also the calibration to be the result of

an estimation process. Then two distances can be modeled the following way. The true distances are $\tilde{d}_1$ and $\tilde{d}_2$. The bias introduced by the miscalibration is $\underline{b}$ with standard deviation σ_b, common to both distances. The noise is $\underline{n}_1$ and $\underline{n}_2$ with common standard deviation σ_n. Calibration and noise are assumed to be independent with mean 0, thus also the bias has expectation 0. We have:

$$\underline{d}_1 = \tilde{d}_1 + \underline{b} + \underline{n}_1 \tag{25}$$
$$\underline{d}_2 = \tilde{d}_2 + \underline{b} + \underline{n}_2 \tag{26}$$
$$\tag{27}$$

As the variances of $\sigma^2_{d_i} = \sigma^2_b + \sigma^2_n$ and the covariance is $\sigma_{d_1 d_2} = \sigma^2_b$ we have the correlation

$$\rho = \frac{\sigma_{d_1 d_2}}{\sigma_{d_1}\sigma_{d_2}} \tag{28}$$
$$= \frac{\sigma^2_b}{\sigma^2_b + \sigma^2_n} \tag{29}$$
$$= \frac{1}{1 + \frac{\sigma^2_n}{\sigma^2_b}} \tag{30}$$

This correlation may be severe if the bias is much larger than the precision of the noise standard deviation. E. g. $\sigma_n = 1$ mm and $\sigma_b = 3$ mm would lead to $\rho = 0.9$ thus 90 % correlation.

This has severe effects on subsequent steps. Assume average distances

$$d = \sum_{i=1}^{n} d_i / n \tag{31}$$

and differences

$$\Delta = d_2 - d_1 \tag{32}$$

to be evaluated. Their standard deviations depend on the correlation:

$$\sigma_d = \sqrt{\frac{1 + (n-1)\rho}{n}}\,\sigma_n \tag{33}$$
$$\sigma_\Delta = \sqrt{2(1-\rho)}\,\sigma_n \tag{34}$$

From (33) and (34) we can draw the following conclusions:

- Averaging of correlated observations only has limited effect as $\lim_{n\to\infty} \sigma_d = \sqrt{\rho}\,\sigma_n$. E. g. 90 % correlation would limit the standard deviation of the mean to be larger than $0.95\sigma_n$. Obviously already moderate correlations severely limit the effect of averaging onto the precision.
- The standard deviation of the difference is significantly smaller for correlated than for uncorrelated data. E. g. 90 % correlation leads to a standard deviation of the difference of $0.45\sigma_n$ compared to 1.4 σ_n for uncorrelated observations, which is a factor 3 in standard deviation or a factor 9 in weight!
- Testing, thus also performance evaluation, essentially depends on the standard deviations of the values to be tested. High correlations lead to misinterpretations in both directions. Testing mean values leads to too optimistic results, whereas testing differences leads to too pessimistic results.

As correlations larger than 90 % or 95 % often occur in vision tasks they should be taken care of in order to avoid wrong conclusions. Or, turning the argument, misleading quality measures may be caused by neglected correlations.

Using *tolerances for reasoning with uncertainty* leads to difficulties. When combining tolerances by determining maximal errors the uncertainty of the mean increases linearly

with the number of observations, whereas the standard deviation only decreases with the square root, which is more realistic. Moreover, correlations cannot easily be incorporated when using tolerances for representing uncertainty. However, in case tolerances are first reduced to standard deviations, e. g. by using the relation $\sigma_x = t_x/k(\alpha)$, then rigorous error propagation is performed, leading to a standard deviation, say σ_y, and finally transformed back to a tolerance by $t_y = k(\alpha)\cdot\sigma_y$, then realistic tolerances are obtained.

4.3.4 Modeling background

Modeling background is a severe problem in vision. Background here is understood to be everything not relevant to solving the specific task, e. g. vegetation when extracting buildings or patterns on wallpaper when navigating in a room.

In some cases it may be necessary to model much more than necessary in order to be safe in evaluating the results. Often, however, partial modeling is sufficient. Long tailed distributions for the position of image features or probabilities of spurious features are a classical tool for modeling background in matching, leading to robust estimators for parameters or cost functions in heuristic search.

In all cases the internal quality measures should at least indicate severe deviations from the underlying assumption, which may be caused by background.

5 Measures are not comparable

5.1 Pro

Many existing vision algorithms only provide ad hoc measures for their evaluation. A typical example is Pratt's measure for describing the performance of edge detection. Measures easily derivable for one algorithm may not be derivable for another one. E. g. some edge detection algorithms easily may be characterized by the standard deviation of the edge position, which is difficult to determine for those which optimize detectability and vice versa. Finally, quality measures may not be related to observable quantities, e. g. fuzzy measures. All these situations make a coherent performance characterization of systems composed of several algorithms difficult if not impossible.

5.2 Contra: Use statistical measures

Following the reasoning of the previous sections, quality measures for characterizing performance must be predictable within the model and at the same time comparable to reality.

Therefore it seems to be reasonable only to use statistically motivated quality measures, such as expectations, variances, probabilities, correlations etc. Other measures, as e. g. fuzzy measures, in general do not allow a link to experiments which is not the case for statistical measures, which often can be related to relative frequencies. In case procedures are not motivated statistically it is worthwhile to reinterpret the decisive measures values statistically in order to have interpretable and testable quality measures at hand.

5.3 Examples

5.3.1 Statistical interpretation of regularization

Regularization is often realized by minimizing a functional, which for reconstructing a function f from observed data g may be of the form,

$$\Omega = \sum_i (f_i - g_i)^2 + \lambda \sum_i \kappa^2(f_i) \tag{35}$$

Here g_i are given observations, f_i are unknown values, thus the first term leads to optimize the fit. The second term, containing the curvatures $\kappa(f_i) = f_{i-1} - 2f_i + f_{i+1}$, contains a penalty for too rough function values f_i. The factor λ can be used to balance both terms, low λ allowing for better fit, high λ enforcing smoother f_i. Classical comments on the choice of λ are: 'We found λ lying in a range 10-20 yielding the best results. The result is not very sensitive to changes in λ'.

Rewriting (35) as

$$\Omega' = \sum_i \left(\frac{f_i - g_i}{\sigma_g} \right)^2 + \sum_i \left(\frac{\kappa(f_i)}{\sigma_\kappa} \right) \tag{36}$$

reveals the regularization term to be the ratio of two variances

$$\lambda = \frac{\sigma_g^2}{\sigma_\kappa^2} \tag{37}$$

This not only gives an intermediate interpretation of λ but allows to avoid any ad-hoc choice by using the precision σ_g of the observed values and the roughness σ_κ of the reconstructed function to determine λ. As both variances can be estimated from real data using variance component estimation (cf. [Brügelmann and Förstner 1992]), *no* free parameter is necessary in regularization. Of course, making the variances dependent on the position allows adaptation to any type of irregularity (cf. [Weidner 1994]).

Moreover, the regularization term suggests the profile to follow an autoregressive scheme of order 2, namely $f_{i+1} = 2f_i - f_{i-1} + \varepsilon_i$ with $\sigma_{\varepsilon_i} = \sigma_\kappa$, allowing to generalize the regularization in case the function actually does not follow this special stochastic process.

Finally, the regularization may be interpreted as Bayesian estimation with the curvature $\kappa \sim N(0, \sigma_\kappa^2)$ as prior information for the f_i, indicating that this procedure can be further generalized.

The reinterpretation of the non-stochastic optimization problem thus not only leads to insight into the semantics of the free parameter but to the elimination of this parameter and to clear hints how to generalize.

5.3.2 Dependency of covariance matrices on the coordinate system

Even if rigorous statistical tools are applied measures may not be comparable. We describe such a pitfall, as it gives insight into the structure of many geometric problems. It occurs in case the result is described in coordinates without explicit reference to the chosen coordinate system: In spite of the same coordinates being given, their variances are not comparable as they are given in different reference systems. The solution goes back to [Baarda 1973] but [Smith 1987a] have independently identified and solved the problem in the context of robotics (cf. also [Smith 1987b]).

Assume a robot is able to measure the length of its path between two positions, while walking on a straight line. Starting at position $P_1(x_1)$ it moves to point $P_2(x_2)$ and then to position $P_3(x_3)$. The measured distances $s_{12} = x_2 - x_1$ and $s_{23} = x_3 - x_2$ are assumed to be independent and have equal standard deviation σ. We now want to express the uncertainty of the complete situation by using the covariance matrix of the vector $\boldsymbol{x}^T = (x_1, x_2, x_3)^T$.

We can argue in at least two ways.

1. Assume position $P_1(x_1)$ to be error free, thus $\sigma_{x_1} = 0$. Then using

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ s_{12} \\ s_{23} \end{pmatrix} = \boldsymbol{A}\boldsymbol{y} \tag{38}$$

by error propagation we obtain the covariance matrix

$$\boldsymbol{\Sigma}_{xx}^{(1)} = \sigma^2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} = \sigma^2 \boldsymbol{A}\boldsymbol{\Sigma}_{yy}\boldsymbol{A}^T \tag{39}$$

2. If we however refer to $P_3(x_3)$ as error free point from which we want to do further reasoning we obtain

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_{12} \\ s_{23} \\ x_3 \end{pmatrix} = \boldsymbol{B}\boldsymbol{z} \tag{40}$$

and thus the covariance matrix

$$\boldsymbol{\Sigma}_{xx}^{(3)} = \sigma^2 \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \sigma^2 \boldsymbol{B}\boldsymbol{\Sigma}_{zz}\boldsymbol{B}^T \tag{41}$$

Observe, we did not change the values of the coordinates but just referred to a different point as being the reference point for the determination of the uncertainty. Both covariance matrices are singular with rank deficiency 1, reflecting the degree of freedom in choosing the origin for the position along the x-axis. It does not seem possible to simply compare variances based on these covariance matrices. The reason is that they refer to different coordinate systems, indicated as suffix.

In spite of both covariance matrices looking different they represent the full information on the geometric configuration as *all* observable quantities derivable from the three coordinates, i. e. all coordinate differences or second differences will have variances independent on whether they are derived from $\boldsymbol{\Sigma}_{xx}^{(1)}$ or $\boldsymbol{\Sigma}_{xx}^{(3)}$. This suggests that the uncertainty of the *form* of the configuration is correctly captured and independent on the chosen coordinate system.

In order to be able to compare the precision of two results possibly given in two different coordinate systems we need to transform the covariance matrices such that they refer to the same coordinate system. This transformation is called an $S - Transformation$, S standing for similarity, indicating that no change in form is intended.

In general it is given by

$$\boldsymbol{\Sigma}_{xx}^{(a)} = \boldsymbol{S}^{(a)}\boldsymbol{\Sigma}_{xx}^{(b)}\boldsymbol{S}^{T(a)} \tag{42}$$

with arbitrary (b) and the projection matrix

$$\boldsymbol{S}^{(a)} = \boldsymbol{I} - \boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{W}^{(a)}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{W}^{(a)} \tag{43}$$

in which $\boldsymbol{H}$ specifies the Jacobian of the coordinate transformation and $\boldsymbol{W}^{(a)}$ specifies the weight of the individual coordinates for defining the coordinate system.

In our special case we have

$$d\boldsymbol{x}^{(1)} = d\boldsymbol{x}^{(3)} + \boldsymbol{H}\,dh = d\boldsymbol{x}^{(3)} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} dh \tag{44}$$

thus the coordinates are just shifted by a differential amount dh, $\boldsymbol{H} = (1\ 1\ 1)^T$. With the weight matrix

$$\boldsymbol{W}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{45}$$

specifying P_1 being the reference point this yields the S-matrix

$$\boldsymbol{S}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \tag{46}$$

and allows to transform the covariance matrix $\boldsymbol{\Sigma}_{xx}^{(3)}$ into $\boldsymbol{\Sigma}_{xx}^{(1)}$ by

$$\boldsymbol{\Sigma}_{xx}^{(1)} = \boldsymbol{S}^{(1)} \boldsymbol{\Sigma}_{xx}^{(3)} \boldsymbol{S}^{T(1)} \tag{47}$$

Choosing $\boldsymbol{W} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ would lead to a S-matrix allowing to transform into system (3).

As $\boldsymbol{S}^{(a)}$ is a projection matrix, a covariance matrix given in *any* coordinate system can be transformed into system (a). In all cases the resultant covariance matrix will have the appropriate rank deficiency.

Generalizations: For a two-dimensional point field with 4 degrees of freedom (translation, rotation, scale) $\boldsymbol{H}$ reads:

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 & x_1 & y_1 \\ 0 & 1 & -y_1 & x_1 \\ 1 & 0 & x_2 & y_2 \\ 0 & 1 & -y_2 & x_2 \\ . & . & . & . \end{pmatrix} \tag{48}$$

where the entries are the given point coordinates.

This example suggests further generalizations. It is essential for the statistical analysis of objects represented in coordinates, which are non-measurable quantities. It confirms the distinction made in invariant theory between *measurable* form- or *shape*-parameters and *non-measurable* so-called *datum* parameters. They specify the reference system in which the coordinates are expressed. Their number and type is fixed by the degrees of freedom of the transformation the object may pass.

6 No theory for algorithms

6.1 Pro

Existing algorithms have shown to work, but are not necessarily based on a theory. Even if they are based on a theory the preconditions are not met, so no use can be made of the theoretical basis.

A classical behavior is: 'My idea is good and works on the examples'.

A new algorithms anyway requires a new setup of data structures, new testing, so, why establish a theory, which in most cases cannot be expected to be general enough to cover unforeseen difficulties.

Anyway, the customer needs a quick solution and making a theory first takes too much time.

6.2 Contra: Performance prediction stimulates theoretical research

There is a tradeoff between quick and dirty solutions and solutions where the theory is worked out, which may require longer development.

But the algorithms should be transferable to more than one application in order to be more efficient. Without theory no prediction on performance is possible: how can one be sure that the algorithm works on the examples on which it was not tested?

Therefore, one definitely should prefer algorithms with a theoretical basis, even if they *seem* suboptimal. This holds for all aspects of algorithms, quality, provability, transparency of behavior, efficiency, algorithmic complexity, etc.

The tradeoff between computing time and quality of result should be made predictable. This seems to be easily feasible for algorithms working on an image pyramid or algorithms using simulated annealing. Especially in real time environments such a tunable performance is of utmost importance in order to exploit the computer resources. Though this requirement is old, nearly no algorithm can be made faster easily with specified loss in performance.

There are many algorithms which have proven to work on a large number of images. There will be reasons for this behavior. Therefore it is advantageous to analyze existing good algorithms in order to understand reasons for their performance. In many cases this analysis will not only lead to clear explanations but also to – possibly significant – improvements.

Finally, formal links between theories need to be established, in order to simplify this type of analysis. An example has been given in section 5.3.1 where deterministically described regularization was linked to statistics, an approach which generalizes to *all* problems which include regularization, as it can easily be related to Bayesian estimation, linking observations and prior information in a well-defined manner.

Obviously, all these recommendations support theoretical research which can be used to advantage for improving the understanding of algorithmic solutions.

7 Too many tuning parameters

7.1 Pro

Many algorithms may have a lot of tuning parameters, making evaluation very difficult. Especially image processing software often contains dozens of routines with tuning parameters being in no way coherent (cf. above). Selecting tuning parameters therefore requires adaption to specific tasks and may need expert knowledge. There are attempts to develop expert systems to find optimal sets of tuning parameters.

7.2 Contra: Only accept meaningful tuning parameters

Testing time obviously grows *exponentially* with the number of tuning parameters. A sequence of n procedures with p parameters each with a domain of d possible values requires the selection of an admissible set out of d^{np} possible parameter values.

A clear consequence is to reduce tuning parameters to a minimum. This would also help automated systems for parameter selection (cf. [Liedtke *et al.* 1993]) As goal/task driven control of algorithms still is necessary, a small set of tuning parameters may be left. But only tuning parameters with a well-defined meaning for control should be allowed. Examples are a significance level for deriving thresholds, or object dependent measures such as the diameter [m] of the object to be detected.

Modularization helps, if constraints can be formulated which reduce the admissible domain in subsequent steps. But this actually is theoretical knowledge!

An example for eliminating a control parameter (λ) in regularization has been given above. Thresholds usually indicate a hypothesis test to take place which may be specified by a significance level. Noise dependency of thresholding can be avoided by using noise estimation techniques. In all cases any type of modeling may reduce the number of tuning parameters or make them semantically meaningful. E. g. extracting features, namely points, lines and regions may be performed with less than a hand-full of parameters,

including a significance level used for *all* tests and an integration scale specifying the expected width of the edges (cf. [Förstner 1994b]).

It seems not to be meaningful to adapt the parameters to the structure of an image unless this structure is representative for a complete class of images, e. g. if the edges are more blurred due to properties of the optics the scale parameters may be increased accordingly. On the other hand, attempts to estimate this parameter locally exactly follow the recommendation to eliminate tuning parameters by a generic model within which the tuning parameter can be estimated from the data.

8 Ground truth is too expensive

8.1 Pro

A severe objection against empirical testing is the difficulty in obtaining ground truth.

In the worst case the objects of interest may not be defined well, e. g. a 'true' segmentation does not exist. This makes empirical testing obsolete.

Ground truth may be expensive, if not too expensive. Paying $ 20 000 for testing the calibration of a robot may appear to be prohibitive.

Even if one would have 1000 examples for empirically testing the algorithms reality does not guarantee the 1001st to be of the same nature, making the effort of empirical testing questionable.

8.2 Contra: Share costs

Yes, empirical testing is expensive. But without empirical testing the customer will not accept a system or a module. Only a mixture of theoretical and empirical evaluation enables predictability of performance and acceptance. Empirical tests tune the parameters, e. g. the noise variance, the likelihood of occlusions or other parameters of the theory, which then can be used to predict the performance.

In order to reduce testing costs *standardization of vision modules* or vision tasks is necessary. This includes the definition of the input/output relation as well as the required performance measures which should be provided by the systems designer.

As empirical tests are expensive *joint tests* are necessary. They allow to exploit the resources of several institutions, academia and industry, in order to define and perform the tests, including the preparation of ground truth, the necessary calibration of the systems, the huge amount of repeated measurements and the proper analysis. Only joint efforts in empirical testing will make vision algorithms acceptable to users.

In order to reduce costs simulated data should be used (cf. below).

8.3 Example: Costs for testing orientation software

An example demonstrates the close interaction between theoretical development and empirical testing on one hand and the activity of academia and the support of users on the other hand.

In the early 70's, a number of Photogrammetric software packages for the simultaneous orientation of large sets of aerial images were developed. In a tutorial on these developments Kraus ([Kraus 1973] reported on 30 projects with between 350 and 7140 unknown points in object space, total 52250. They were manually checked in the field. The total cost per point was estimated to be in the range between $ 12 and $ 15, including, flying costs, film development, manual measurements, computing costs, and field work. The projects were performed within two years. The survey agencies strongly supported the university as there was an interest in the results. The reason for the interest was motivated by the theoretical research going ahead ([Ackermann 1966]) or parallel indicating the power of the proposed methods.

9 Simulations are not reality

9.1 Pro

Simulations cannot replace reality, they always are too ideal. As generating artificial images uses the same model as the analysis no statement can be made on the behavior under real circumstances. One is not able to model all types of nasty disturbances occuring when confronting an algorithm with real data.

9.2 Contra: Simulations replace complicated theory

All the arguments are correct. However, most algorithms are based on some theoretical framework, and implementation and theory are always different. Examples are discretization of continuous models, unavailability of theoretical tools for analyzing the behavior of algorithms, suboptimal implementation for which no theory is available, etc. ... last, but not least: coding errors.

Moreover, simulated images can to some extent replace real images in order to reduce the costs for establishing ground truth.

Therefore simulations seem to be unavoidable in order to

- prove the correctness of implementations
- analyze the behavior of algorithms under varying conditions
- develop performance measures.

They support theoretical analyses where analytical tools are not powerful enough.

Of course, simulations cannot serve as a surrogate for real experiments, which are necessary to tune the models to reality.

9.3 Example: Evaluating segmentation results

We want to discuss the problem of evaluating segmentation results, segmentation being a key problem in image analysis.

For this purpose we assume the segmentation to yield lists of basic features $\widehat{\mathsf{F}}$, namely points $\widehat{\mathsf{P}}$, lines $\widehat{\mathsf{L}}$ and regions $\widehat{\mathsf{R}}$, and relations between all these features. Features and relations can be assumed to be attributed.

As for real images no true segmentation is available we propose to evaluate segmentations based on simulated inputs, where the true segmentation containing the true features F is known.

For analyzing the relation between given and estimated features one can follow Fuchs ([Fuchs *et al.* 1994]) and build up a transition table $\boldsymbol{T} = (t_{ij})$ indicating whether given and estimated features meet. For determining the relation $\textsc{meet}(\widehat{\mathsf{F}}_i, \mathsf{F}_j)$ we use the exoskeleton leading to areas $\mathcal{A}(\mathsf{F}_j)$ and $\mathcal{A}(\widehat{\mathsf{F}}_i)$ around each feature and determine t_{ij} by

$$t_{ij} = \begin{cases} 0 & \text{if } \widehat{\mathsf{F}}_i \cap \mathcal{A}(\mathsf{F}_j) = \emptyset \text{ and } \mathsf{F}_j \cap \mathcal{A}(\widehat{\mathsf{F}}_i) = \emptyset \\ 1 & \end{cases} \tag{49}$$

Thus, $t_{ij} = 1$ if a given feature is close to an estimated one. The distance threshold hereby is defined by the skeleton. $t_{ij} = 0$ definitely excludes any type of closeness.

The sums

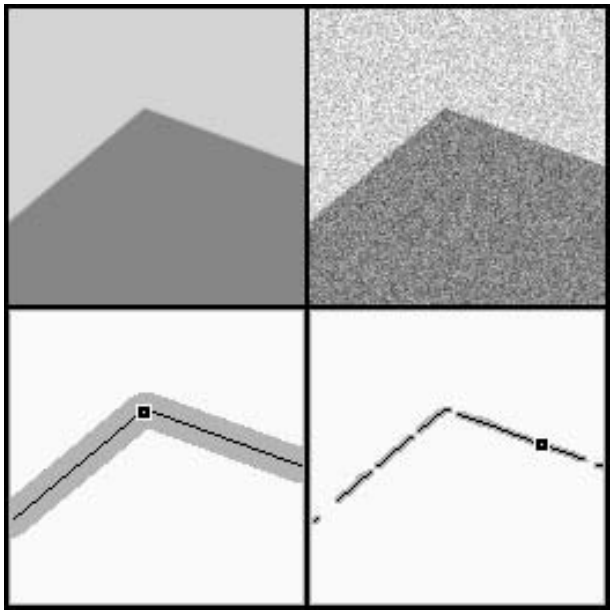
$$p_j = \sum_i t_{ij} \quad \text{and} \tag{50}$$

$$m_i = \sum_j t_{ij} \tag{51}$$

have a very definite meaning:

1. p_j measures the *degree of partitioning*

Figure 2: The 120° corner image with noise $\sigma_n^2 = 75\,[\mathrm{gr}^2]$, the 'ideal' features F and the 'extracted' features $\widehat{\mathsf{F}}$ derived by feature extraction. The white parts correspond to the regions, the black pixel chains correspond to the linear features, points are indicated by black squares. No attempt has been made to optimize the quality of the result (from FUCHS et al. 1994).



$p_j \geq 2$: a given feature F_j is partitioned into p_i features
$p_j = 1$: a feature F_j is not partitioned
$p_j = 0$: a feature F_j is lost.

2. m_i measures the *degree of merging*

$m_i \geq 2$: m_i given features are merged into one estimated $\widehat{\mathsf{F}}_i$
$m_i = 1$: the estimated $\widehat{\mathsf{F}}_i$ feature is not a merging of several given features
$m_i = 0$: the estimated feature $\widehat{\mathsf{F}}_i$ is spurious.

An example of such a transition matrix for F_j and $\widehat{\mathsf{F}}_i$ shown in Fig. 2 is given in Table 1. The degrees p_j and m_i for partitioning and merging are given for each individual feature type and all features. The given point P_a was lost, the estimated point $\widehat{\mathsf{P}}_1$ is spurious among the set of points. The edges L_b (right) and L_c (left) both are splitted. The two regions R_d and R_e have been merged into one estimated region $\widehat{\mathsf{R}}_9$. The off diagonal parts of the table indicate transitions from one feature type into another, e. g. the point P_a has been 'changed' into the edge $\widehat{\mathsf{L}}_2$.

From this comparison, which can be fully automated, a number of performance measures for characterizing the segmentation can be derived:

- Ocurrance of features and relations

	P_a	$m(\mathsf{P})$	L_b	L_c	$m(\mathsf{L})$	R_d	R_e	$m(\mathsf{R})$	$m(\mathsf{F})$
$\widehat{\mathsf{P}}_1$	0	**0**	1	0	**1**	0	0	**0**	**1**
$p(\widehat{\mathsf{P}})$	**0**		**1**	**0**		**0**	**0**		
$\widehat{\mathsf{L}}_2$	1	**1**	0	1	**1**	0	0	**0**	**2**
$\widehat{\mathsf{L}}_3$	0	**0**	1	0	**1**	0	0	**0**	**1**
$\widehat{\mathsf{L}}_4$	0	**0**	0	1	**1**	0	0	**0**	**1**
$\widehat{\mathsf{L}}_5$	0	**0**	1	0	**1**	0	0	**0**	**1**
$\widehat{\mathsf{L}}_6$	0	**0**	1	0	**1**	0	0	**0**	**1**
$\widehat{\mathsf{L}}_7$	0	**0**	0	1	**1**	0	0	**0**	**1**
$\widehat{\mathsf{L}}_8$	0	**0**	0	1	**1**	0	0	**0**	**1**
$p(\widehat{\mathsf{L}})$	**1**		**3**	**4**		**0**	**0**		
$\widehat{\mathsf{R}}_9$	0	**0**	0	0	**0**	1	1	**2**	**2**
$p(\widehat{\mathsf{R}})$	**0**		**0**	**0**		**1**	**1**		
$p(\widehat{\mathsf{F}})$	**1**		**4**	**4**		**1**	**1**		

Table 1: Transition table for the example shown in Fig. 2.

  - Probability of a given point to be found
  - Probability of a point-line incidence to be found
  - Probability of a point-region incidence to be found
  - Probability of two regions on the left and right side of a given edge to merge
- The quality of edge extraction
  - Probability of an edge pixel to be found leading to the above mentioned coverage
  - The average length of the edge segments replacing a long edge.
  - The average number of edges sitting on a given edge, giving the degree of partitioning.
  - The average number of points erroneously sitting on edges
- Spurious Features
  - The average number of spurious points per image area
  - The average number of spurious edges per image area
  - The length distribution of spurious edges

It is to be discussed whether a formal definition of a segmentation output together with its quality measures can be found as basis for future empirical comparisons.

10 Testing is not acknowledged

10.1 Pro

Testing takes time. Following a rule of thumb one can use the time relation *theory : implementation : testing = 1 : 10 : 100*. Also the relation *working algorithm : published working algorithm = 1 : 10*, not counting the costs for getting ground truth. These efforts would not demotivate doing experimental work, if it were acknowledged. But reading the

calls for high-level international conferences, one realizes that *new* ideas seem to be worth more than tested old ones. The effort to get a paper accepted is certainly lower (cf. above) when writing a theoretical paper including a new idea than when reporting on an extensive empirical (well done) investigation. As 99 % of the research is carried out by PhD students it is understandable that their first intent is to finish their thesis, rather than replicating ideas of others and showing their deficiencies. It seems still to be the psychological barrier of not being acknowledged which hinders doing the hard work of establishing procedures with well-documented performance characteristics.

10.2 Contra: Empirical testing is worthwhile

It is difficult to change this situation, as it is not only a technical problem which is to be solved.

A few arguments should give hints about how to reevaluate development of well-designed vision algorithms: The lifetime of an algorithm is proportional to testing time perhaps even to the square of the testing time.

It should be more satisfying having developed a procedure which is used after having finished the thesis, than knowing the work will never be used.

Thorough analysis of algorithms, which includes theoretical studies and empirical testing, improves understanding, i. e. without doubt is of scientific value.

Therefore reimplementation of algorithms for clarifying their potential should be strongly supported. Of course this requires much better documentation, but in most cases – as a side effect – leads to clearer algorithms. (Long) papers on vision algorithms should be evaluated with respect to the degree the reader can verify the results based on the available information. This may be simplified by providing at least the test data (images) and the code or by offering to run the algorithms on data provided by the reader. The communication techniques are available.

It remains to repeat the old requirement: increase acceptance of providing performance measures and empirical testing by supervisors, funding agencies, editors, ...

11 Conclusions

The 10 most common type of arguments against investing work into performance characterization and empirical testing have been discussed. Though each of them is true to some extent, promoting a research field and transferring its result to real world application cannot bypass the long tradition in engineering science where quality evaluation of products is a common tool. The examples given are meant to indicate that many tools for characterizing the quality of vision algorithms are already available, some of course need to be refined or developed.

Even if not all types of problems can be rigorously analyzed to full satisfaction, at least the basic tools in image understanding should be analyzed rigorously with respect to the different aspects of application. But also theoretical work together with representative examples for its use will be necessary in order to come to a commonly accepted methodology of performance characterization in Computer Vision.

References

[Ackermann 1966] Ackermann, F. (1966): On the Theoretical Accuracy of Planimetric Block Triangulation. *Photogrammetria*, 21:145–170, 1966.

[Baarda 1973] Baarda, W. (1973): *S-Transformations and Criterion Matrices*, Band 5 der Reihe 1. Netherlands Geodetic Commission, 1973.

[Brügelmann and Förstner 1992] Brügelmann, R.; Förstner, W. (1992): Noise Estimation for Color Edge Extraction. In: Förstner, W.; Ruwiedel, S. (Eds.), *Robust Computer Vision*, pages 90–107. Wichmann, Karlsruhe, 1992.

[Förstner 1987] Förstner, W. (1987): Reliability Analysis of Parameter Estimation in Linear Models with Applications to Mensuration Problems in Computer Vision. *Computer Vision, Graphics & Image Processing*, 40:273–310, 1987.

[Förstner 1992] Förstner, W. (1992): "Uncertain Spatial Relationships and their Use for Object Location in Digital Images". In: Förstner W., Haralick R. M., Radig B. (Ed.), *Robust Computer Vision - Tutorial Notes*. Institut für Photogrammetrie, Universität Bonn, 1992.

[Förstner 1994a] Förstner, W. (1994): Diagnostics and Performance Evaluation in Computer Vision. In: *Proc. Performance versus Methodology in Computer Vision, NSF/ARPA Workshop, Seattle*, pages 11–25. IEEE Computer Society, 1994.

[Förstner 1994b] Förstner, W. (1994): *A Framework for Low Level Feature Extraction*, pages 383–394. LNCS 802. Springer, 1994.

[Fuchs *et al.* 1994] Fuchs, C.; Lang, F.; Förstner, W. (1994): "On the Noise and Scale Behaviour of Relational Descriptions". In: *Int. Arch. f. Photogr. and Remote Sensing*, Band XXX, 1994.

[Haralick 1985] Haralick, R. M. (1985): Computer Vision Theory: The Lack Thereof. In: Shapiro, L..; Kak, A. (Ed.), *Proccedings of the Third Workshop on Computer Vision: Representation and Control*, pages 113–121. IEEE CS, 1985.

[Jain 1991] Jain, R. C.; Binford, T. O. (1991): Ignorance, Myopia, and Naivité in Computer Vision Systems. *CVGIP: Image Understanding*, 53(1):112–117, 1991.

[Kraus 1973] Kraus, K. (1973): Die Katasterphotogrammetrie im praktischen Einsatz. In: F., Ackermann (Ed.), *Numerische Photogrammetrie*, Sammlung Wichmann, Neue Folge. Wichmann Karlsruhe, 1973.

[Liedtke *et al.* 1993] Liedtke, C.-E.; Schnier, Th.; Bloemer, A. (1993): Automated Learning of Rules Using Genetic Operators. In: *Proc. ICAP 93*, 1993.

[Price 1985] Price, K. (1985): I've Seen Your Demo; So What? In: Shapiro, L.; Kak, A. (Ed.), *Proceedings of Third Workshop on Computer Vision: Representation and Control*, pages 122–124. IEEE CS, 1985.

[Smith 1987a] Smith, R., Self M. Cheeseman P. (1987): Estimating Uncertain Spatial Relationships in Robotics. In: *Uncertainty in Artificial Intelligence, Vol. 2*. North Holland, 1987.

[Smith 1987b] Smith, R., Self M. Cheeseman P. (1987): A Stochastic Map for Uncertain Spatial Relationships,. In: *Symposium on Robotics Research*. MIT Press, 1987.

[Weidner 1994] Weidner, U. (1994): Parameterfree Information-Preserving Surface Restoration. In: Eklundh, J.-O. (Ed.), *Computer Vision - ECCV 94, Vol. II, Proceedings*, pages 218–224, 1994.