Plane Detection in Point Cloud Data

Michael Ying Yang, Wolfgang Förstner

michaelyangying@uni-bonn.de, wf@ipb.uni-bonn.de

TR-IGG-P-2010-01

January 25, 2010

Technical Report Nr. 1, 2010

Department of Photogrammetry
Institute of Geodesy and Geoinformation
University of Bonn

Available at
http://www.ipb.uni-bonn.de/technicalreports/

Plane Detection in Point Cloud Data

Michael Ying Yang

michaelyangying@uni-bonn.de

Abstract

Plane detection is a prerequisite to a wide variety of vision tasks. RANdom SAmple Consensus (RANSAC) algorithm is widely used for plane detection in point cloud data. Minimum description length (MDL) principle is used to deal with several competing hypothesis. This paper presents a new approach to the plane detection by integrating RANSAC and MDL. The method could avoid detecting wrong planes due to the complex geometry of the 3D data. The paper tests the performance of proposed method on both synthetic and real data.

1.1 Introduction

Due to their abundance in man-made environments, as well as to their attractive geometric properties, planes are commonly used in various vision tasks. As reported in the literature, planes have been successfully employed in diverse applications such as grouping (Van Gool et al., 1998), 3D reconstruction and scene analysis (Kaucic et al., 2001; Kahler and Denzler, 2006), object recognition (Rothwell et al., 1995), segmentation (Biosca and Lerma, 2008), and augmented reality (Simon et al., 2000). A plane segmentation of point clouds based on fuzzy clustering methods is proposed in (Biosca and Lerma, 2008). (Kahler and Denzler, 2006) presents a method to detect physically present 3D planes in scenes imaged with a handheld camera. (Poppinga et al., 2008) proposes a fast plane detection and polygonalization in noisy 3D range images. (Leonardis et al., 1997) finds shapes by concurrently growing different seed primitives from which a suitable subset is selected according to minimum description length (MDL) criterion.

In computer vision, one of the most widely known methodologies for plane detection is the RANdom SAmple Consensus (RANSAC) algorithm (Fischler and Bolles, 1981). It has been proven to successfully detect planes in 2D as well as 3D. RANSAC is reliable even in the presence of a high proportion of outliers. Its principle is well explained by (Fischler and Bolles, 1981;

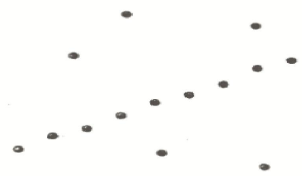

Figure 1.1: 14 points in a plane: 9 points on a straight line and 5 outliers?

McGlone et al., 2004). In the field of automatic buildings modeling based on Lidar data, many authors suggest its use for achieving different tasks. For example, (Brenner et al., 2001) use RANSAC algorithm for detecting the building roof planes. (Schnabel et al., 2007) use RANSAC to detect basic shapes, such as planes, spheres, cylinders, cones, in point clouds. In our case, RANSAC algorithm is used with the aim of plane detection.

The following paper presents a new approach to the plane detection in point cloud data by integrating RANSAC and MDL. In section 1.2 we first introduce the principle of MDL encoding using a simple example for interpreting a set of points in a plane. In section 1.3 we derive the description length of interpreting points in 3D space as a generalization of section 1.2. Section 1.4 gives the basic approach of RANSAC algorithm for plane detection. Section 1.5 gives the proposed plane detection method by integrating RANSAC and MDL. The experimental result is given in section 1.6, followed by the concluding remarks.

1.2 Interpreting a Set of Points in a Plane

We first introduce the principle of minimum description length (MDL) encoding using a simple example. Let n_0 points x_i, y_i in a plane be given as in Fig. 1.1. The scope is to explain the data in the most intuitive manner. This figure suggests the larger number $n = 9$ of the $n_0 = 14$ points to approximately sit on a straight line, while the other $\bar{n} = n_0 - n = 5$ points do not belong to this line. Fig. 1.2 shows a different pattern, where we are not sure whether we should assume the 5 points in the middle of the figure to belong to a straight line or whether we rather should treat the figure as consisting of 14 randomly distributed points or even 3 vertical nearly straight lines.

The situation is representative for a large class of interpretation tasks: 1. We have to deal with several competing hypothesis which have a different structure; 2. We have to deal with a significant amount of spurious data;

Figure 1.2: 14 points in a plane: random set or 5 points on a straight line and 9 outliers?

3. There may be no explanation of the data within the assumed set of hypothesis.

The problem of explaining the data sets in Fig. 1.1 and Fig. 1.2 lies in the fact that the pure fit between a selected number of data points and a set of hypothesized straight lines, say, is not sufficient as a quality measure, as this fit can be made perfect by restricting to just 2 data points or by increasing the number of postulated straight lines. Therefore the evaluation of an explanation has to balance the fit between data and model and the complexity of the model. The principle of description length encoding fulfills these requirements.

We want to derive the description lengths in bits for the case when no model, only outliers, is assumed with the case when the data essentially are assumed to consist of points sitting approximately on a straight line admitting some outliers. Let the coordinates be given up to a resolution of ϵ (e. g. 1 pixel) and be within a range R (e. g. 256 pixel). Then $lb(R/\epsilon)$ bits are necessary to describe one coordinate, here $lb(\cdot) = \log_2(\cdot)$. The description length for the n_0 points, when assuming outliers (O), therefore is

$$\Phi_0 = \#bits(points \mid O) = n_0 \cdot 2lb(R/\epsilon) \tag{1.1}$$

thus $2 \cdot n_0 \cdot 8 = 16n_0$ in the case of no points in a $256 \cdot 256$ pixel image or 224 bits on the plot of Fig. 1.1.

If we now assume n points to sit on a straight line and the other $\bar{n} = n_0 - n$ points to be outliers (1L+O), we need

$$\Phi_1 = \#bits(points \mid 1L + O) = n_0 + \bar{n} \cdot 2lb(R/\epsilon) +$$
$$\left[n \cdot lb(R/\epsilon) + \sum_{i=1}^{n} \left\{ \frac{1}{2ln2} \cdot (\frac{v_i}{\sigma})^2 + lb(\sigma/\epsilon) + \frac{1}{2} lb2\pi \right\} \right] + 2lb(R/\epsilon) \tag{1.2}$$

where the first term represents the n_0 bits for specifying whether a point is good or bad, the second term is the number of bits to describe the bad points, the third term is the number of bits to describe the good points and the last term is needed to describe the 2 parameters of the straight line, which is the number of bits to describe the model complexity, a variation of (Rissanen, 1978). We assumed the good points to randomly sit on the straight line which leads to the first term in the brackets, and to have Gaussian distributed derivations v_i from the line with standard derivation σ. (Förstner, 1989) shows that $\frac{1}{2ln2} \cdot (\frac{x-\mu}{\sigma})^2 + lb(\sigma/\epsilon) + \frac{1}{2}lb2\pi$ bits are necessary to describe a Gaussian variable $x \sim N(\mu, \sigma^2)$, when μ and σ^2 are given and if it is rounded to multiples of ϵ.

In the example of Fig. 1.1, with $n = 9$ and $\bar{n} = 5$ we on an average need:

$$\begin{aligned}\Phi_1 &= n_0 + \bar{n} \cdot 2lb(R/\epsilon) + n\left(lb(R/\epsilon) + lb(\sigma/\epsilon) + \frac{1}{2}lb2\pi\right) + 2lb(R/\epsilon) \\ &= 14 + 5 \cdot 2 \cdot 8 + 9 \cdot (8 + 1 + 2.04) + 2.8 \approx 209bits \quad (1.3)\end{aligned}$$

to code the point set, when assuming a straight line with outliers. This is less than the 224 bits, thus supporting this explanation. For Fig. 1.2 we however need 229 bits, assuming 5 points sitting on a straight line, which obviously is no explanation for the data.

1.3 Interpreting a Set of Points in 3D Space

In this section, we want to derive the description length of interpreting points in 3D space. Given a set of points, we assume several competing hypothesis, here namely, outliers (O), 1 plane and outliers (1P+O), 2 planes and outliers (2P+O), 3 planes and outliers (3P+O), ect..

Let n_0 points x_i, y_i, z_i be given in a 3D coordinate and the coordinates be given up to a resolution of ϵ and be within range R. The description length for the n_0 points, when assuming outliers (O), therefore is

$$\Phi_0 = \#bits(points \mid O) = n_0 \cdot (3lb(R/\epsilon)) \quad (1.4)$$

where $lb(R/\epsilon)$ bits are necessary to describe one coordinate.

If we now assume n points to sit on a plane and the other $\bar{n} = n_0 - n$ points to be outliers, we need

$$\begin{aligned}\Phi_1 = \#bits(points \mid 1P + O) &= n_0 + \bar{n} \cdot 3lb(R/\epsilon) + 3lb(R/\epsilon) + n \cdot 2lb(R/\epsilon) \\ &\left[\sum_{i=1}^{n}\left\{\frac{1}{2ln2} \cdot (\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) + \frac{1}{2}lb(|\Sigma|/\epsilon^6) + \frac{k}{2}lb2\pi\right\}\right] \quad (1.5)\end{aligned}$$

where the first term represents the n_0 bits for specifying whether a point is good or bad, the second term is the number of bits to describe the bad points, the third term is the number of bits to describe the 3 parameters of the plane, which is the number of bits to describe the model complexity, a variation of (Rissanen, 1978). We assumed the good points to randomly sit on the plane which leads to the fourth term, and to have Gaussian distribution $x \sim N(\boldsymbol{\mu}, \Sigma)$. We show in Appendix that $\frac{1}{2ln2} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} lb(|\Sigma| / \epsilon^6) + \frac{k}{2} lb 2\pi$ bits are necessary to describe a Gaussian variable $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{x} = (x_1, \cdots, x_k)$, $\boldsymbol{\mu}$ and Σ are given, and if it is rounded to multiples of ϵ.

If we now assume n_1 points to sit on a plane, n_2 points to sit on the second plane, and the other $\bar{n} = n_0 - n_1 - n_2$ points to be outliers, we need

$$\begin{aligned} \Phi_2 = \#bits(points \mid 2P + O) = n_0 + \bar{n} \cdot 3lb(R/\epsilon) + 6lb(R/\epsilon) \\ + n_1 \cdot 2lb(R/\epsilon) + n_2 \cdot 2lb(R/\epsilon) \\ \left[\sum_{i=1}^{n_1+n_2} \left\{ \frac{1}{2ln2} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} lb(|\Sigma| / \epsilon^6) + \frac{k}{2} lb 2\pi \right\} \right] \end{aligned} \tag{1.6}$$

where the first term represents the n_0 bits for specifying whether a point is good or bad, the second term is the number of bits to describe the bad points, the third term is the number of bits to describe the parameters of two planes. We assumed the n_1 good points to randomly sit on one plane which leads to the fourth term, and the n_2 good points to randomly sit on the other plane which leads to the fifth term, and to have Gaussian distribution $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \Sigma)$ which leads to the sixth term.

Similarly, assume n_1 points to sit on a plane, n_2 points to sit on the second plane, n_3 points to sit on the third plane, and the other $\bar{n} = n_0 - n_1 - n_2 - n_3$ points to be outliers, we need

$$\begin{aligned} \Phi_3 = \#bits(points \mid 3P + O) = n_0 + \bar{n} \cdot 3lb(R/\epsilon) + 9lb(R/\epsilon) \\ + n_1 \cdot 2lb(R/\epsilon) + n_2 \cdot 2lb(R/\epsilon) + n_3 \cdot 2lb(R/\epsilon) \\ \left[\sum_{i=1}^{n_1+n_2+n_3} \left\{ \frac{1}{2ln2} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} lb(|\Sigma| / \epsilon^6) + \frac{k}{2} lb 2\pi \right\} \right] \end{aligned} \tag{1.7}$$

where the first term represents the n_0 bits for specifying whether a point is good or bad, the second term is the number of bits to describe the bad points, the third term is the number of bits to describe the parameters of three planes. We assumed the n_1, n_2, n_3 good points to randomly sit on respective planes which leads to the fourth, fifth, and sixth terms, and to have Gaussian distribution $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \Sigma)$ which leads to the seventh term.

This procedure can generalize to other shape primitives, such as sphere, cylinder, cone, ect..

1.4 RANSAC Algorithm for Plane Detection

The principle of RANSAC algorithm consists to search the best plane among a 3D point cloud. In the same time, it reduces the number of iterations, even if the number of points is very large. For this purpose, it selects randomly three points and it calculates the parameters of the corresponding plane. Then it detects all points of the original cloud belonging to the calculated plane, according to a given threshold. Afterwards, it repeats these procedures N times; in each one, it compares the obtained result with the last saved one. If the new result is better, then it replaces the saved result by the new one.

This algorithm needs four input data which are:

- The 3D point cloud (*point-list*) which is a matrix of three coordinate columns X, Y and Z;
- The tolerance threshold of distance t between the chosen plane and the other points. Its value is related to the altimetric accuracy of the point cloud;
- The *forseeable-support* is the maximum probable number of points belonging to the same plane. It is deduced from the point density and the maximum foreseeable roof plane surface.
- The probability α is a minimum probability of finding at least one good set of observations in N trials. It lies usually between 0.90 and 0.99.

Algorithm 1 details the pseudocode of RANSAC algorithm.

In Algorithm 1, ϵ is a percentage of observations allowed to be erroneous; the function *pts2plane* calculates the plane parameters from three chosen points. The function *dist2plane* calculates the signed distances between point set and given plane.

1.5 Proposed Plane Detection Algorithm

In this section, we show how we detect planes in 3D point cloud. RANSAC is applied to extract planes. The basic idea is to estimate the model parameters using the minimum number of data possible and then to check which of the remaining data points fit the model estimated, as shown in section 1.4.

Based on the observation that RANSAC may find wrong planes if the data has a complex geometry, we use the following scheme for plane extraction:

- The point cloud is partitioned into small rectangular blocks to make sure that there will be a maximum of three planes in one block.

Algorithm 1 RANSAC for plane detection

$bestSupport = 0$; $bestPlane(3,1) = [0, 0, 0]$
$bestStd = \infty$; $i = 0$
$\epsilon = 1$ - *forseeable-support*/length(*point-list*)
$N = \text{round}(log(1 - \alpha)/log(1 - (1 - \epsilon)^3))$
while $i \leq N$ **do**
  j = pick 3 points randomly among (*point-list*)
  pl = pts2plane(j)
  dis = dist2plane(pl, *point-list*)
  $s = \text{find}(abs(dis) \leq t)$
  st = Standard-deviation(s)
  if (length(s) $>$ *bestSupport*) or (length(s) $=$ *bestSupport* and $st <$ *bestStd*) **then**
    bestSupport = length(s)
    bestPlane = pl; *bestStd* = st
  end if
  $i = i + 1$
end while

- RANSAC is applied to extract planes in each block.
- The MDL principle is employed to decide how many planes are in each block. Eventually, there are zero to three planes in each block.

Further, we can apply region growing to merge the neighboring planes within certain local range. Geometric features are then extracted for interpreting man-made objects (Schmittwilken et al., 2009).

Algorithm 2 details the pseudocode of above proposed algorithm.

1.6 Experimental Results

In this section, we illustrate the results of the proposed plane detection method obtained with some synthetic and real data. The real-world entrance stair data is acquired from a terrestrial laser scanner. The synthetic building facade data is derived from the attribute grammar by a random based derivation (Schmittwilken et al., 2009).

We have applied our plane detection method to real 3D range data of an entrance stair. We present some of the results of a data set of about 16473 points (cf. Fig. 1.3). Fig. 1.4 shows the results of three planes detected by our method. The points belonging to the detected plane are removed, as

Algorithm 2 Proposed algorithm for plane detection

partition point cloud into rectangular blocks
Assume: a maximum of three planes in each block
Initialize: $\Phi_0, \Phi_1, \Phi_2, \Phi_3$
for each block **do**
    calculate Φ_0, as in eq. 1.4
    apply RANSAC 1 to extract a plane
    calculate Φ_1, as in eq. 1.5
    remove the points belonging to the first plane
    apply RANSAC to extract a plane
    calculate Φ_2, as in eq. 1.6
    remove the points belonging to the second plane
    apply RANSAC to extract a plane
    calculate Φ_3, as in eq. 1.7
    calculate $i^* = \arg_i \max \frac{exp(-\Phi_i)}{\sum_{i=0}^{3} exp(-\Phi_i)}$, i^* planes detected
end for

stated in Algorithm 2. Fig. 1.5 shows the results of two planes detected by our method.

We have also tested our algorithm on several synthetic building facade data sets. We exemplarily present one of the tested synthetic data sets (cf. Fig. 1.6). With original building facade of 0.4 Million points, Fig. 1.6 top left, 2165 planes are detected, Fig. 1.6 top right. The point cloud is divided into nonoverlapping 32×32 rectangular blocks. Fig. 1.6 bottom shows zoom-in planes of a window and the entrance. The plane normal vectors are also shown and colors here are for visualization purpose.

1.7 Conclusion

We propose a new approach to the plane detection in point cloud data. We derive the description length of interpreting points in 3D space, and review the basic RANSAC approach for plane detection. By integrating RANSAC and MDL, the approach could avoid detecting wrong planes due to the complex geometry of the 3D data. The proposed approach shows good performance on both synthetic and real data. Future work will be establishing a unified description length framework of other basic shape primitives, such as sphere, cylinder, and cone.

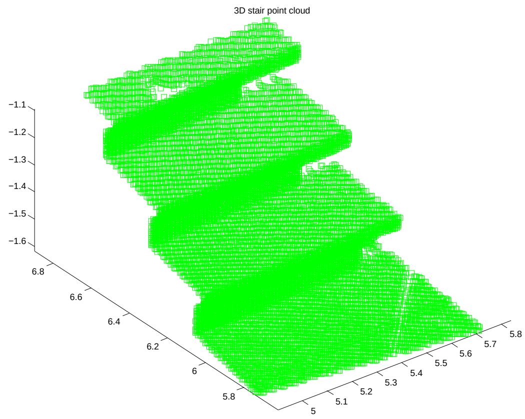

Figure 1.3: Input point cloud of a stair.

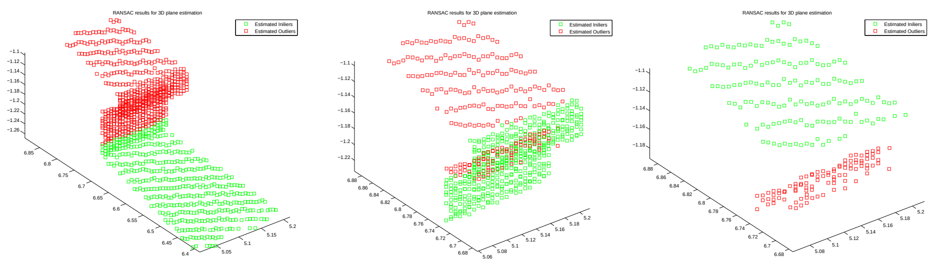

Figure 1.4: **Left**: First plane detected in one block of stair (Fig. 1.3). **Middle**: Second plane detected. **Right**: Third plane detected

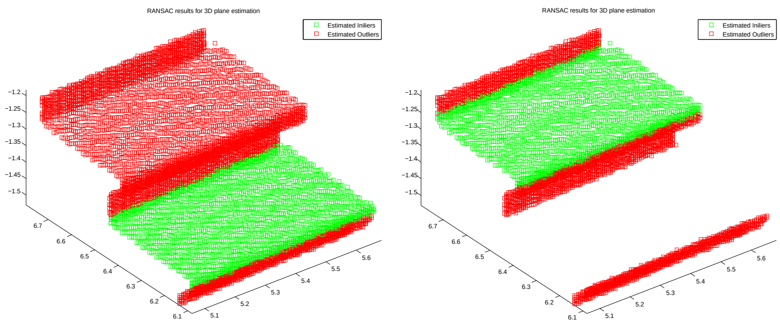

Figure 1.5: **Left**: First plane detected in one block of stair (Fig. 1.3). **Right**: Second plane detected.

Appendix

The theory of information was developed by Shannon (Shannon, 1948) for analyzing communication systems. Specifically it deals with measuring the information content of a message and the efficiency of sending the message over a channel which possibly is noisy. The theory is of a statistical nature as it only is concerned with the statistical properties of the message not with its meaning.

According to Shannon a discrete information source can be modeled as a Markov-Process, which randomly selects letters out of a prespecified alphabet. The information, which is transmitted per letter, is the larger the less likely the letter is selected and can be interpreted as the degree of surprise when the letter reaches the receiver or as the uncertainty when no knowledge about the letter is available. In the most simple case the transmitted letters are independent. Let $P(\underline{a} = w_i)$ be the probability that the letter $\underline{a}$ (a random variable) is equal to the value w_i. Then the gain of information when being told w_i, i. e. the information of w_i is

$$I(\underline{a} = w_i) = I(w_i) = -lbP(w_i) \tag{1.8}$$

The unit of information is 'bit' here.

In a similar manner one can measure the information which is obtained when being told w_i, but already knows the value of another letter $\underline{b} = w_j$.

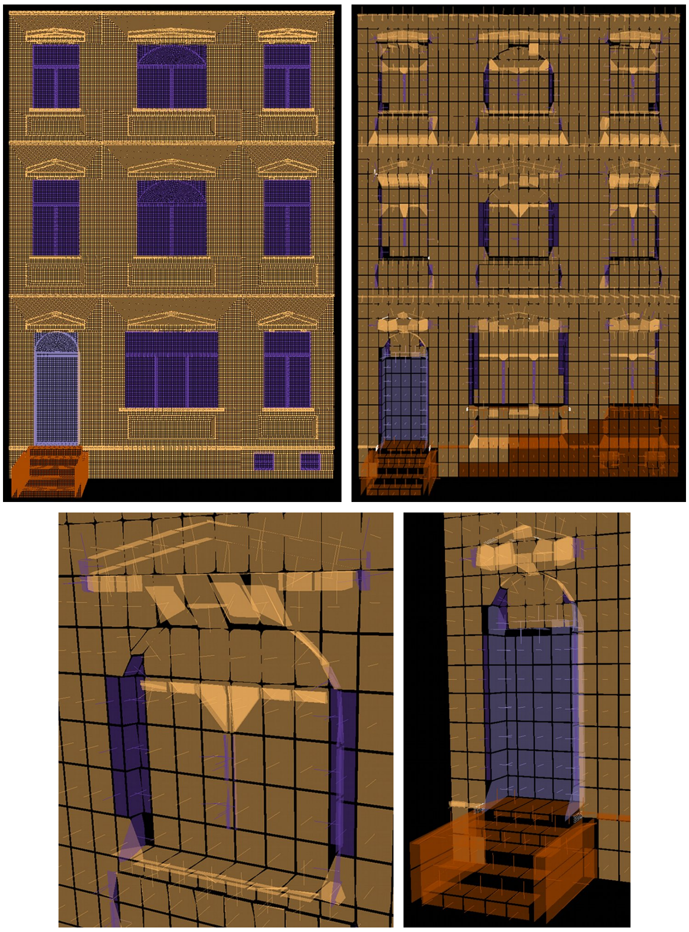

Figure 1.6: **Top Left**: Input point cloud of building facade, with color coded ground truth. **Top Right**: Planes detected of building facade. **Bottom Left**: Zoom-in part of *window* planes. **Bottom Right**: Zoom-in part of *stair* and *door* planes.

With the conditional probability $P(w_i \mid w_j)$ we obtain the conditional information

$$I(w_i \mid w_j) = I(w_i, w_j) - I(w_j) \tag{1.9}$$

In case the events $\underline{a} = w_i$ and $\underline{b} = w_j$ are independent, $P(w_i \mid w_j) = P(w_i)$, the information we obtain is identical to that without preknowledge. If however, the events are dependent the information obtained when being told w_i is smaller than without preknowledge.

Assume a random variable being uniform distribution $\underline{x} \sim U[a, b]$, according to Eq. 1.8, we have

$$I_U(x \mid a, b) = lb(b - a) \tag{1.10}$$

Similar way, assume a random variable being Gaussian distribution $\underline{x} \sim N(\mu, \sigma^2)$, we have

$$I_N(x \mid \mu, \sigma^2) = \frac{1}{2ln2} \cdot (\frac{x - \mu}{\sigma})^2 + \frac{1}{2} lb2\pi\sigma^2 \tag{1.11}$$

If we round a Gaussian random variable $\underline{x}$ to a resolution of ϵ, yielding $\underline{x_r}$, then, using Eq. 1.9,

$$\begin{aligned} I_r(x \mid \mu, \sigma^2, \epsilon) &= I_N(x \mid \mu, \sigma^2) - I_U(-\frac{\epsilon}{2}, \frac{\epsilon}{2}) \\ &= \frac{1}{2ln2} \cdot (\frac{x - \mu}{\sigma})^2 + lb(\sigma/\epsilon) + \frac{1}{2} lb2\pi \end{aligned} \tag{1.12}$$

Observe that I_r only represents the bits necessary to code the difference, $x - \mu$, to the mean, as we have assumed the mean, the precision, and the resolution to be known. As could be expected, minimizing the number of bits when presenting a random number corresponds to only state the necessary digits with respect to its precision.

Assume a random variable being Gaussian distribution $\underline{\boldsymbol{x}} \sim N(\mu, \Sigma)$, we have

$$I_N(\boldsymbol{x} \mid \mu, \Sigma) = \frac{1}{2ln2} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} lb(|\Sigma|) + \frac{k}{2} lb2\pi \tag{1.13}$$

where $\boldsymbol{x} = (x_1, \cdots, x_k)$.

If we round a Gaussian random variable $\underline{\boldsymbol{x}}$ to a resolution of ϵ, yielding $\underline{\boldsymbol{x}_r}$, then, using Eq. 1.9,

$$\begin{aligned} I_r(\boldsymbol{x} \mid \mu, \Sigma, \epsilon) &= I_N(\boldsymbol{x} \mid \mu, \Sigma) - k \cdot I_U(-\frac{\epsilon}{2}, \frac{\epsilon}{2}) \\ &= \frac{1}{2ln2} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} lb(|\Sigma| \, \epsilon^{2k}) + \frac{k}{2} lb2\pi \end{aligned} \tag{1.14}$$

Bibliography

Biosca, J. and J. Lerma (2008). Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing 63*(1), 84–98.

Brenner, C., N. Haala, and D. Fritsch (2001). Towards fully automated 3D city model generation. In *In Automatic Extraction of Man-Made Objects from Aerial and Space Images III*.

Fischler, M. and R. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM 24*, 381–395.

Förstner, W. (1989). Image analysis techniques for digital photogrammetry. In *Photogrammetrische Woche 1989*, pp. 205–221.

Kahler, O. and J. Denzler (2006). Detection of planar patches in handheld image sequences. In *PCV06*, pp. xx–yy.

Kaucic, R., R. Hartley, and N. Dano (2001). Plane-based projective reconstruction. *Computer Vision, IEEE International Conference on 1*, 420.

Leonardis, A., A. Jaklic, and F. Solina (1997). Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 1289–1295.

McGlone, J. C., E. M. Mikhail, and J. Bethel (Eds.) (2004). *Manual of Photogrammetry (5th edition)*. ASPRS.

Poppinga, J., N. Vaskevicius, A. Birk, and K. Pathak (2008). Fast plane detection and polygonalization in noisy 3d range images. In *International Conference on Intelligent Robots and Systems (IROS)*, pp. 3378–3383.

Rissanen, J. (1978). Modelling by shortest data description. In *Automatica*, Volume 14, pp. 465–471.

Rothwell, C. A., A. Zisserman, D. A. Forsyth, and J. L. Mundy (1995). Planar object recognition using projective shape representation. *Int. J. Comput. Vision 16*(1), 57–99.

Schmittwilken, J., M. Y. Yang, W. Förstner, and L. Plümer (2009). Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects. *Annals of GIS 15*(2), 117–126.

Schnabel, R., R. Wahl, and R. Klein (2007). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*(2), 214–226.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal 27*, 379–423, 623–656.

Simon, G., A. W. Fitzgibbon, and A. Zisserman (2000). Markerless tracking using planar structures in the scene. *Augmented Reality, International Symposium on 0*, 120.

Van Gool, L., M. Proesmans, and A. Zisserman (1998). Planar homologies as a basis for grouping and recognition. *IVC 16*(1), 21–26.