Discrete plane segmentation and estimation from a point cloud using local geometric patterns

Yukiko Kenmochi, Lilian Buzer, Akihiro Sugimoto, Ikuko Shimizu

▶ **To cite this version:**

Yukiko Kenmochi, Lilian Buzer, Akihiro Sugimoto, Ikuko Shimizu. Discrete plane segmentation and estimation from a point cloud using local geometric patterns. International Journal of Automation and Computing, 2008, 5 (3), pp.246-256. 10.1007/s11633-008-0246-1 . hal-00622344

HAL Id: hal-00622344
https://hal.science/hal-00622344

Submitted on 20 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrete plane segmentation and estimation from a point cloud using local geometric patterns

Yukiko KENMOCHI[1,*] Lilian BUZER[1] Akihiro SUGIMOTO[1,2] Ikuko SHIMIZU[3]

[1] Université Paris-Est, Institut Gaspard-Monge, CNRS/UMLV/ESIEE - Laboratoire A2SI, ESIEE
Cité Descartes, BP 99, 93162 Noisy-le-Grand Cedex, France

[2] National Institute of Informatics, Chiyoda, Tokyo 101-8430, Japan

[3] Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganedai, Tokyo 184-8588 Japan

Abstract: This paper presents a method for segmenting a 3D point cloud into planar surfaces using recently obtained discrete-geometry results. In discrete geometry, a discrete plane is defined as a set of grid points lying between two parallel planes with a small distance, called thickness. Contrarily to the continuous case, there exist a finite number of local geometric patterns (LGPs) appearing on discrete planes. Moreover, such a LGP does not possess the unique normal vector but a set of normal vectors. By using those LGP properties, we first reject non-linear points from a point cloud, and then classify non-rejected points whose LGPs can have common normal vectors into a planar-surface-point set. From each segmented point set, we also estimate parameters of a discrete plane by minimizing its thickness.

Keywords: Image segmentation, discrete geometry, planar surface.

1 Introduction

Recent progress of computer vision technologies allows us to easily acquire a 3D point cloud of an object[1]. Let us consider a simple case where our interesting object is polyhedral. Then, reconstructing the whole 3D shape using several 3D point clouds taken from different viewpoints requires extracting at least three common planar patches from every 3D point cloud[2]. This extraction is known as surface segmentation.

Conventional approaches for the surface segmentation problem of a 3D point cloud are classified into the following three categories: region-based, edge-based, and hybrid approaches. The first one merges points that have similar region properties calculated from their neighboring points such as normal vectors [3, 4], curvatures [5] parameters of fitted planes [6, 7, 8] or quadratic surfaces [6, 9], and other indices corresponding to local surface shapes [10]. As calculated properties are very sensitive to noise and quantization errors, they cause over segmentation results [11]. Thus, an additional procedure for merging regions is needed after the initial segmentation [11, 12]. In the second approach, edges are searched such that they separate regions by using depth discontinuities [13]. As edges are not always extracted as connected curves, they cause under segmentation, contrary to the first one. Thus, in this case, an additional procedure for splitting regions is needed after the initial segmentation. The third approach is hybrid between the above two approaches, namely, combinations of region-based and edge-based approaches [11, 14, 15]. One of interesting ideas for planar cases in the third approach can be found in [14]; the notion of locally planar points are proposed for a planar segmentation method. Locally planar points are used for detecting not only planar regions but also edges, because points that are not locally planar are considered to be potentially edge points.

The above three approaches possess the common problem of using surface primitives or geometric features for the surface segmentation. In order to approximate/select surface primitives and calculate geometric features from 3D discrete points, we are obligated to set parameters from the practical point of view. Such parameter setting/adaptation is not simple work, because it depends on the discreteness of a given 3D point cloud, such as data resolution and noise. For example, we need to define a set of neighboring points for calculating geometric features for each 3D point. Note that, in this paper, a set of neighboring points in a 3D point cloud is also called a locally geometric pattern, abbreviated to an LGP. The sizes and patterns of LGPs implicitly give influences to other parameter values in the post-process of region merging/splitting, because calculated geometric features generally have some errors due to various patterns of LGPs. However, in most cases, such parameter adaption is realized experimentally or statistically with some statistical hypothesis. In this paper, simplifying the surface segmentation problem by focusing only on planar cases, we present a new method for planar surface segmentation of a 3D point cloud using recently obtained discrete-geometry results [16]. The discrete geometry enables us to eliminate many unnecessary parameters and to simplify the planar surface segmentation algorithm.

We present a discrete version of the hybrid method, in this paper, by using fixed-size LGPs in a discrete space. As a consequence, once we fix the size, it automatically decides other parameter values thanks to the theory of discrete geometry [16]. In discrete geometry, a discrete plane is defined as a set of grid points lying between two parallel planes with a small distance, called a thickness [16]. Contrarily to the continuous case, there exist a finite number of local geomet-

*Corresponding author. E-mail address: y.kenmochi@esiee.fr; Address: Laboratoire A2SI, ESIEE, Cité Descartes, BP 99, 93162 Noisy-le-Grand Cedex, France; Fax: +33.1.45.92.66.99

ric patterns, LGPs, appearing on discrete planes, called linear LGP [17]. In fact, points whose LGPs are linear can be considered to be discrete version of locally planar points [14]. In addition, each linear LGP does not possess the unique normal vector but a set of normal vectors [18]. By using those LGP properties, we present a segmentation method following the two steps: first reject non-linear points from a point cloud (edge-based part), and then merge non-rejected points whose LGPs have common normal vectors (region-based part). It thus uses only precalculated look-up tables with respect to LGPs, and does not require any parameter setting. Furthermore, our method is less sensitive to noise as well as quantization errors. Indeed linear LGPs already take into account quantization errors for their generation. We show the effectiveness by applying our algorithm to 3D point clouds such as range images. In order to evaluate our segmentation results, we apply a method for estimating discrete plane parameters from each segmented planar surface by minimizing its thickness. This problem is solved by a linear programming method. As the thickness indicates the segmentation inaccuracy, we consider that the thinner the thickness, the better the segmentation result.

2 Non-linear point rejection using LGP

2.1 Discrete planes

Let $\mathbb{R}$ be the set of real numbers. A plane $\mathbf{P}$ in the 3D Euclidean space $\mathbb{R}^3$ is defined by the following expression:

$$\mathbf{P} = \{(p,q,r) \in \mathbb{R}^3 \ : \ \alpha p + \beta q + \gamma r + \delta = 0\}$$

where $\alpha, \beta, \gamma, \delta \in \mathbb{R}$. Let $\mathbb{Z}^3$ be the set of grid points whose coordinates are integers in $\mathbb{R}^3$. A discrete plane, which is a digitization of $\mathbf{P}$, is then defined such that

$$\mathbf{D}(\mathbf{P}) = \{(p,q,r) \in \mathbb{Z}^3 : 0 \leq \alpha p + \beta q + \gamma r + \delta < \omega\} \quad (1)$$

where $\omega = \max(|\alpha|, |\beta|, |\gamma|)$, called the thickness [16].

2.2 Linear LGP on discrete planes

We consider a cubical grid-point set $\mathbf{Q}(\boldsymbol{x})$ whose edge length is 2 around a point $\boldsymbol{x} \in \mathbb{Z}^3$ such that

$$\mathbf{Q}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{Z}^3 : \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq 1\}. \quad (2)$$

Let us assume that each point in $\mathbb{Z}^3$ has a binary value such as either 1 or 0. Such a pattern of binary points in $\mathbf{Q}(\boldsymbol{x})$ is called local geometric patterns, abbreviated to LGP. There are 2^{26} different LGP for $\mathbf{Q}(\boldsymbol{x})$ providing that the central point $\boldsymbol{x}$ always has the fixed value 1. This indicates that $\boldsymbol{x}$ is considered not to be a background point but to be a surface point.

Among those different LGPs, we investigated which LGP can appear on discrete planes [17]. This problem is mathematically written as follows. Let $\mathbf{F}$ be a set of points whose binary values are 1 in $\mathbf{Q}(\boldsymbol{x})$. If there is a plane $\mathbf{P}$ such that

$$\begin{aligned}\mathbf{F} &= \mathbf{D}(\mathbf{P}) \cap \mathbf{Q}(\boldsymbol{x}) \\ &= \{(p,q,r) \in \mathbf{Q}(\boldsymbol{x}) : 0 \leq \alpha p + \beta q + \gamma r + \delta < \omega\}, \quad (3)\end{aligned}$$

we say that $\mathbf{F}$ forms a discrete plane in $\mathbf{Q}(\boldsymbol{x})$. Therefore, our problem is solved by looking for all possible $\mathbf{F}$, namely

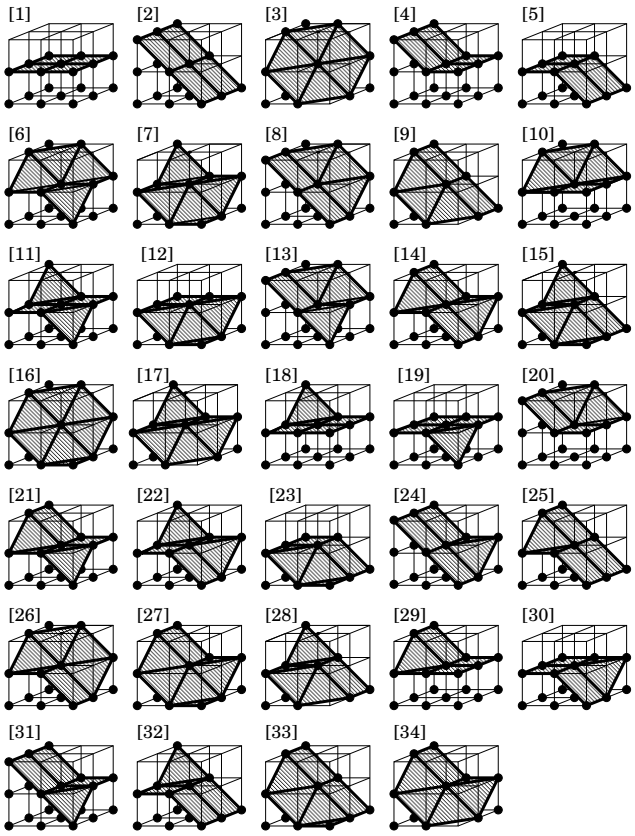


Figure 1 The 34 linear LGPs.

LGPs, satisfying (3). Such LGPs are called linear LGPs. Since this problem is considered to be the feasibility of the inequalities of (3) for all $(p,q,r) \in \mathbf{F}$, we need to check if there are feasible solutions $\alpha, \beta, \gamma, \delta$ for each different LGP of $\mathbf{Q}(\boldsymbol{x})$. If they exist, such LGP can appear on discrete planes and become linear LGP.

However, [17] shows that we can avoid computing the feasibility test for all 2^{26} LGPs of $\mathbf{Q}(\boldsymbol{x})$, by taking an approach based on arithmetic planes [16, 20], which are related to discrete planes. An algorithm is then proposed to generate all linear LGPs, and it is found that there exist only 34 LGPs that appear on discrete planes, called linear LGPs, up to translations, rotations and symmetries, as shown in Fig. 1. Note that they are generated with the constraints

$$0 \leq \alpha \leq \beta \leq 1, \ \gamma = 1. \quad (4)$$

In order to visualize the shapes of linear LGPs in Fig. 1, we add polyhedral meshes generated for planar surface points by applying a discrete-marching-cube-like method for the 18-neighborhood system [21] to a digitized half space. Interior points of planar surfaces are designated as black points in the figures.

2.3 Locally linear and non-linear points

Experimentally, those linear LGPs can be seen not only on discrete planes but also on discrete smooth surfaces. Intuitively, this is not difficult to understand, since any local surface patch on a smooth surface can be approximated to

a planar surface when the size of the patch becomes small. In the discrete space, even if a point has a linear LGP, we are uncertain whether such a point appears on a planar surface or a non-planar surface. Contrarily, if a point has a non-linear LGP, it never appears on a planar surface. From this reason, if a point has a linear LGP, it is called a locally linear point, otherwise, simply called a non-linear point.

2.4 From a point cloud to a grid point set

Before executing the non-linear point rejection to a grid-point set, we explain how to transform a 3D point cloud into a grid-point set. Our input in this paper is a range image represented by a 2D digital image each of whose pixel $(x, y) \in [X_1, X_2] \times [Y_1, Y_2]$ of $\mathbb{Z}^2$ has a depth information $d(x, y)$ from a 3D scanner to an object surface. We transform such a range image into a 3D triple-valued image by re-quantizing a depth $d(x, y)$ as follows: for each point (x, y, z) in a finite subset $\mathbf{X} = [X_1, X_2] \times [Y_1, Y_2] \times [Z_1, Z_2]$ of $\mathbb{Z}^3$, we define a triple-valued function such that

$$t(x,y,z) = \begin{cases} 2, & \text{if } z = \lfloor \frac{d(x,y)}{r} + \frac{1}{2} \rfloor, \\ 1, & \text{if } z > \lfloor \frac{d(x,y)}{r} + \frac{1}{2} \rfloor, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where r is a sampling interval for depths.

Grid points whose values are 2 are closest to input points $(x, y, d(x, y))$ so that they are considered to be discrete surface points and to be visible from a 3D scanner. Thus, we call them visible surface points and define a set of visible surface points such that

$$\mathbf{V} = \{(x,y,z) \in \mathbf{X} \ : \ t(x,y,z) = 2\}. \tag{6}$$

Concerning grid points whose values are 1, they are invisible from a 3D scanner so that we do not know whether they are surface points or not. Therefore, we simply call them invisible points. Since the rest of grid points whose values are 0 are visible and background points, a set of potential points for an object is defined as a union of visible surface points and invisible points such that

$$\mathbf{W} = \{(x,y,z) \in \mathbf{X} \ : \ t(x,y,z) \neq 0\}.$$

Thus, a set of surface points is obtained as a border point set of $\mathbf{W}$ such that

$$\partial\mathbf{W} = \{\boldsymbol{x} \in \mathbf{X} \ : \ \mathbf{N}_6(\boldsymbol{x}) \cap \overline{\mathbf{W}} \neq \emptyset\} \tag{7}$$

where

$$\mathbf{N}_6(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{Z}^3 \ \|\boldsymbol{x} - \boldsymbol{y}\|_1 \leq 1\}$$

and $\overline{\mathbf{W}}$ is the complement of $\mathbf{W}$. Note that $\mathbf{V} \subseteq \partial\mathbf{W}$ and the equality does not always hold.

A visible surface point set $\mathbf{V}$ can be considered to be a digitization of a point cloud, while a surface point set $\partial\mathbf{W}$ is necessary for making binary patterns of LGPs; the binary value of a point $\boldsymbol{x}$ is set to be 1 if $\boldsymbol{x} \in \partial\mathbf{W}$; otherwise, set to be 0. This is why we also need $\partial\mathbf{W}$ as well as $\mathbf{V}$.

2.5 Non-linear point rejection

By simply checking the LGP linearity, we can therefore reject non-linear points from a grid-point set, since we know that non-linear points never appear on any discrete plane.

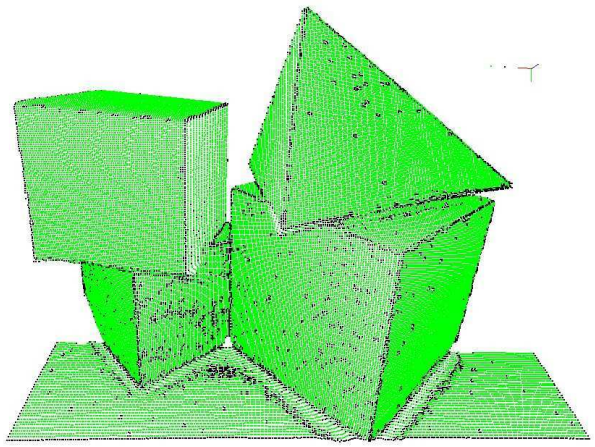


Figure 2 An experimental example of non-linear point rejection.

In other words, the linear LGPs play an important role in filtering linear points. Note that it is realized by looking up the binary table of LGPs (linear or not).

For the experiment, we use a 3D point cloud taken by a 3D scanner Konica-Minolta VIVID 910 with a resolution 320×240. We first re-quantized the z-coordinates with a similar interval r to those of the x- and y-coordinates from (5), and obtained two finite grid-point sets, namely, a visible surface point set $\mathbf{V}$ and a surface point set $\partial\mathbf{W}$, from (6) and (7). Note that the LGP linearity is checked for every point in $\mathbf{V}$ even if binary patterns for LGPs are made from $\partial\mathbf{W}$.

Figure 2 shows an example of locally linear and non-linear points, colored in light green and black respectively, in a 3D point cloud taken by a 3D scanner Konica-Minolta VIVID 910 with a resolution 320×240. We see in the figure that points appearing around polyhedral-face edges are rejected as well as isolated points that are considered to be noise. However, we also observe that some points around edges are not rejected, because they are considered to be locally linear even if they are not linear in a larger region than their LGPs. This fact implies that a simple post-processing, such as the connected component labeling [16] of a non-rejected point set, does not always give satisfactory results for planar surface segmentation.

In fact, we can generalize the definition of a cubical grid-point set $\mathbf{Q}(\boldsymbol{x})$ with an infinity norm that is not more than k, instead of 1, in (2) [17]. If $k = 2$, for example, we obtain 1574 linear LGPs. However, larger LGPs are not so useful for the non-linear point rejection. First, they are more sensitive to noise because each point need more neighboring points to be locally linear. Therefore, we generally obtain more black points in Fig. 2 if we use larger LGPs. Secondly, from the practical point of view, we lose a privilege to use a binary look-up table for checking the LGP linearity, because of the size of all binary patterns of LGPs. In the cases where k is more than 1, we need to use another data structure such as a tree to store all linear LGPs and to check the linearity of a given LGP. Because of these reasons, we use LGPs for $k = 1$ in this paper.

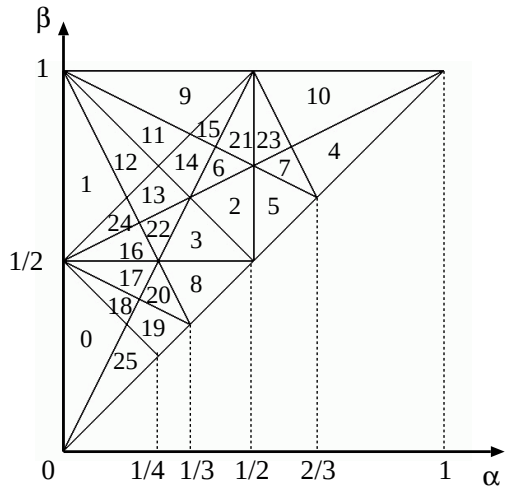


Figure 3 Normal cells on the $\alpha\beta$-plane with constraint (4).

3 Planar surface segmentation of locally linear points

In order to solve our segmentation problem, we propose a method using not only the point connectedness but normal vectors derived from LGPs.

3.1 Feasible normal vectors of linear LGPs

A linear LGP is a discrete plane patch of $\mathbf{D}(\mathbf{P})$ in a bounded space $\mathbf{Q}(\boldsymbol{x})$, denoted by $\mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$. Given a $\mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$, we can find a set of Euclidean planes $\mathbf{P}$ such that the digitization of each of those planes in $\mathbf{Q}(\boldsymbol{x})$ is equal to $\mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$. The set of all such Euclidean planes is called the preimage and it is known that the correspondence between discrete plane patches and Euclidean planes is not one-to-one but one-to-many [18]. Because of the one-to-many correspondence, the preimage of $\mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$ is represented by a set of parameters $\alpha, \beta, \gamma, \delta$. More precisely, the preimage is obtained as a feasible solution set of the inequality set of (3) for all points $(p, q, r) \in \mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$. It means that the preimage is given by a convex polytope in the parameter space [18].

As all interesting parameters in this paper are translation-invariant, we focus on the three parameters α, β, γ indicating the normal vector of $\mathbf{P}$, distinguished from the intercept δ of $\mathbf{P}$. We thus apply the Fourier-Motzkin elimination [22] to the inequality set of (3) for all $(p, q, r) \in \mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$, so that a set of feasible normal vectors is calculated from each linear LGP. Remark that all calculations are done by using only integers, i.e., they cause no rounding errors; the details are found in [17].

The results are derived in the space (α, β) from linear LGP with the constraints (4), since we use the 34 linear LGPs in Fig. 1. The feasible region for each linear LGP is obtained as a convex polygon in the triangle region whose vertices are $(0, 0)$, $(0, 1)$ and $(1, 1)$ of the space (α, β) because of (4). Each line in the triangle region in Fig. 3 corresponds to a half plane represented by each inequality of (3) for every $(p, q, r) \in \mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$ for every linear LGP. We see in Fig. 3 that the inequality set divides the triangle region into triangular or quadrilateral polygons in the space (α, β), called normal cells. The feasible region of each linear LGP is given as a set of normal cells that constitutes a convex polygon in the space (α, β). Table 1 shows the set of normal cells whose union corresponds to the convex polygon representing the set of feasible normal vectors for each linear LGP depicted in Fig. 1. Remark that there are some pairs of linear LGPs both of which have the identical set of normal cells. In addition, a normal cell corresponds not only to a simple linear LGP but to several linear LGPs. Thus, the correspondence between linear LGPs and normal cells is many-to-many.

Table 1 Linear LGPs and their normal cells.

linear LGP	normal cells
1	0 25
2	1 9 11 12
3	4 5 7 10 23
4,5	0 1 16 17 18 24
6,17	2 3 4 5 7 8
7	2 3 5 8
8,9	6 9 10 11 14 15 21 23
10,12	8 19 20 25
11	8 17 18 19 20
13,28	2 3 4 5 6 7 9 10 11 12 13 14 15 21 22 23
14	2 3 6 13 14 15 16 21 22 24
15	2 3 6 11 12 13 14 22
16	4 5 7 10 23
18,19	0 18 19 25
20,23	0 1 3 8 12 13 16 17 18 19 20 22 24 25
21,22	3 8 16 17 20 22
24,25	1 9 11 12 13 14 15 24
26,34	2 4 5 6 7 10 21 23
27	2 5 6 7 21 23
29,30	0 17 18 19 20 25
31,32	1 12 13 16 22 24
33	6 9 11 14 15 21

3.2 Discrete Gaussian sphere

The 26 normal cells in Fig. 3 are generated with the constraints (4). We embed these normal cells into the 3D space (α, β, γ) with $\gamma = 1$, as illustrated in Fig. 4. The triangle surrounded by thick lines in Fig. 4 corresponds to the triangular region that is the union of normal cells in Fig. 3. Once the normal cells are embedded into the space (α, β, γ), we make the congruous ones by applying to them 48 transformations of rotations and symmetries of a cube of edge length 2, centered at the origin of the 3D space. We see, in Fig. 4, that there are the 48 triangles on the cube, so that the whole cube contains 1248 normal cells. Such a cube is called the cubical Gaussian sphere.

We now project normal cells tiled on the cubical Gaussian sphere onto a unit sphere centered at the origin, as illustrated in Fig. 5. The unit sphere separated by projected normal cells is called the discrete Gaussian sphere,

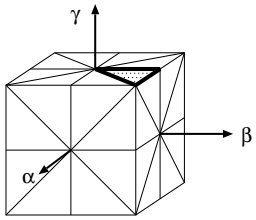


Figure 4 The cubical Gaussian sphere.

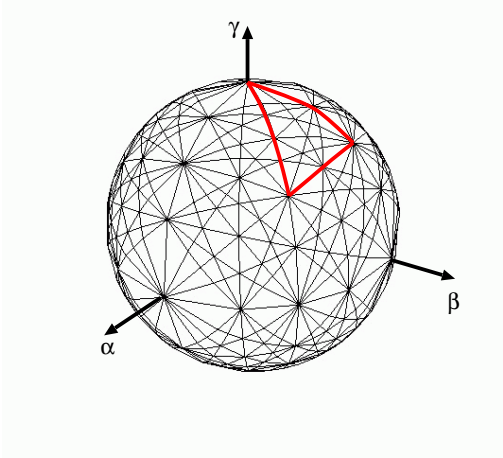


Figure 5 The discrete Gaussian sphere.

because the size of normal cells indicates the resolution of digitized normal vectors calculated from linear LGPs. The triangle surrounded by red lines in Fig. 5 corresponds to the triangle surrounded by thick lines in Fig. 4 that corresponds to the union of normal cells in Fig. 3. In the remainder, we denote $\mathbf{G}$ the set of all normal cells on the discrete Gaussian sphere. Remark that we use only integer or rational numbers to calculate all normal cells, which are related to the cubical Gaussian sphere.

3.3 Unified discrete Gaussian image

By using the discrete Gaussian sphere, we give a discrete version of extended Gaussian images that are useful for representing surface shapes [23], called unified discrete Gaussian images. Let us first consider a discrete version of the Gaussian image that is the mapping from an object surface point to its normal vector on the Gaussian sphere. Let $\mathbf{V}'$ be a locally linear point set in $\mathbb{Z}^3$. For a point $\boldsymbol{x} \in \mathbf{V}'$, we define a discrete Gaussian image $\mathbf{I}(\boldsymbol{x})$ as the set of normal cells corresponding to the linear LGP of $\boldsymbol{x}$. Choosing a normal cell $c \in \mathbf{G}$, we now consider a point subset of $\mathbf{V}'$ such that

$$\mathbf{R}(c) = \{\boldsymbol{x} \in \mathbf{V}' : c \in \mathbf{I}(\boldsymbol{x})\}. \tag{8}$$

We then obtain the number of points in $\mathbf{R}(c)$ for every $c \in \mathbf{G}$, called the unified discrete Gaussian image, such that

$$u(c) = |\mathbf{R}(c)|. \tag{9}$$

Note that $u(c)$ and $\mathbf{R}(c)$ are generated by simply looking up a table such as Table 1.

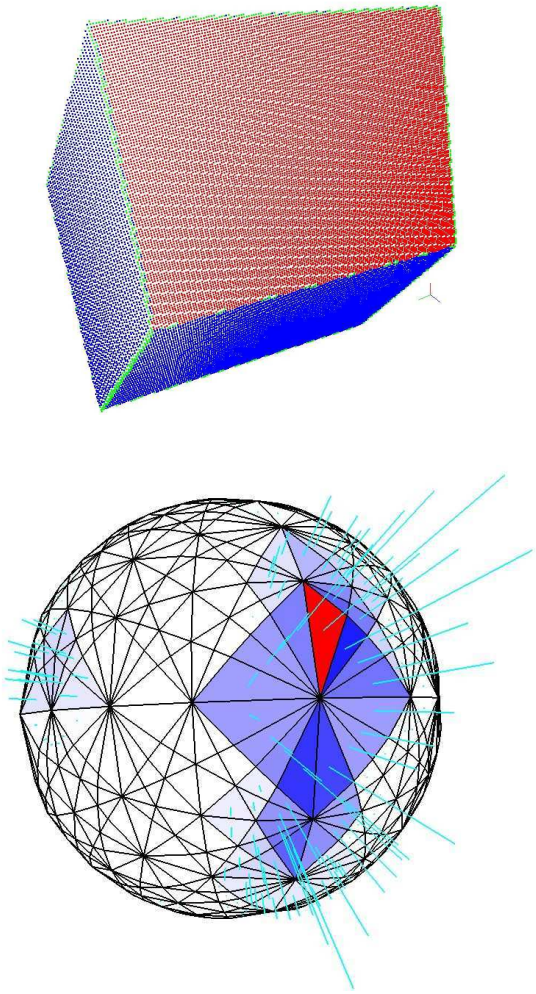


Figure 6 A synthetic 3D image of a box (top) and its unified discrete Gaussian image (bottom).

The concept of unified discrete Gaussian images is similar to that of extended Gaussian images [23]. The differences from extended Gaussian images are the followings: the function (9) is defined with respect to a normal cell c on the discrete Gaussian sphere $\mathbf{G}$, instead of a point $\boldsymbol{n}$ on the Gaussian sphere; the value of (9) is the number of grid points $\boldsymbol{x}$ such that $\mathbf{I}(\boldsymbol{x})$ includes c, instead of the area of the surface whose normal vector is $\boldsymbol{n}$. From the definition, we see that our unified discrete Gaussian image represents a distribution of normal cells of a digital object surface.

Figure 6 shows an example of the unified discrete Gaussian images for a digitized box. Concerning cell colors on the discrete Gaussian sphere in Fig. 6 (bottom), the darker the blue cell, the larger the value of $u(c)$, and the red cell has the maximum value. The length of the pale blue needle for each cell c also corresponds to the value of $u(c)$. On a digitized box in Fig. 6 (top), red and blue points are locally linear, while green points are non-linear. Note that red points correspond to the red cell in Fig. 6 (bottom). Figure 6 shows that we can extract a set of grid points that belong to a digital plane $\mathbf{D}(\mathbf{P})$ by choosing a "correct" normal cell, for example, a red one. This is based on the following fact;

Algorithm 1: Planar surface segmentation

input : a unified discrete Gaussian image $u(c)$, point sets $\mathbf{R}(c)$, and a minimum surface size s
output: planar-surface point sets $\mathbf{S}_i$ for $i = 1, 2, 3, \ldots$

1 **begin**
2 initialize a label such that $l = 0$;
3 **repeat**
4 make a queue D_k of normal cells with priorities of values $u(c)$;
5 increment l and initialize $\mathbf{S}_l = \emptyset$;
6 set h to be the highest priority cell in D_k and remove it from D_k;
7 **while** $|\mathbf{R}(h)| > \max(s-1, |\mathbf{S}_l|)$ **do**
8 set $\mathbf{C}$ to be the maximum connected component of $\mathbf{R}(h)$;
9 **if** $|\mathbf{C}| > |\mathbf{S}_l|$ **then** set $\mathbf{S}_l = \mathbf{C}$;
10 reset h to be the highest priority normal cell in D_k and remove it from D_k;
11 **if** $|\mathbf{S}_l| \geq s$ **then**
12 **forall** c *such that* $u(c) \neq 0$ *and* $\mathbf{R}(c) \cap \mathbf{S}_l \neq \emptyset$ **do**
13 reset $\mathbf{R}(c) = \mathbf{R}(c) \setminus \mathbf{S}_l$ and $u(c) = |\mathbf{R}(c)|$;
14 **until** $|\mathbf{S}_l| < s$;
15 **return** $\mathbf{S}_i$ for $i = 1, 2, \ldots, l-1$;
16 **end**

if (α, β, γ) is a normal vector of $\mathbf{D}(\mathbf{P})$, (α, β, γ) is included in the common normal cell(s) of $\mathbf{I}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbf{D}(\mathbf{P})$.

3.4 Algorithm

By using the unified discrete Gaussian image $u(c)$ and the point sets $\mathbf{R}(c)$, we present our algorithm for planar surface segmentation from a locally linear point set $\mathbf{V}'$. Our problem is formulated as follows; each point $\boldsymbol{x} \in \mathbf{V}'$ is assigned into one of sets $\mathbf{S}_i$ for $i = 1, 2, \ldots$ such that the points in each $\mathbf{S}_i$ constitutes a connected planar-surface set. From the previous discussions, our method is founded on the following hypothesis: if there is a connected point subset $\mathbf{S} \subseteq \mathbf{V}'$ such that they have a common normal cell for all $\boldsymbol{x} \in \mathbf{S}$, $\mathbf{S}$ may constitute a discrete plane.

Based on this hypothesis, we present Algorithm 1. we look for the largest connected grid-point set $\mathbf{S}_i$, whose points having a common normal cell by using $u(c)$ and $\mathbf{R}(c)$. As each point has several normal cells, our method cannot be processed in parallel with respect to normal cells. It must be a repeated procedure; once we obtain $\mathbf{S}_i$, we remove all points of $\mathbf{S}_i$ from every $\mathbf{R}(c)$, modify $u(c)$, and repeat this procedure after the increment of i. Practically, we would like to avoid obtaining a very small surface patch, so that we set a parameter s that is the minimum size for $\mathbf{S}_i$.

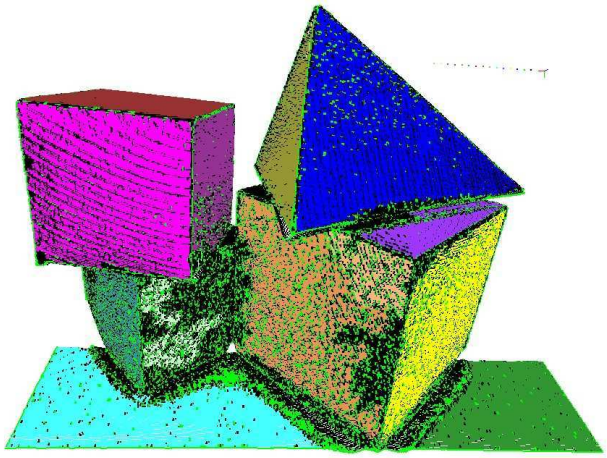
Algorithm 1 is thus a loop procedure of seeking planar surfaces $\mathbf{S}_i$. Each $\mathbf{S}_i$ is a maximally connected point set, whose points have a common normal cell. Once we find $\mathbf{S}_i$, we check the size of $\mathbf{S}_i$ in Step 11, and if $|\mathbf{S}_i| \geq s$, we remove all points of $\mathbf{S}_i$ from every $\mathbf{R}(c)$ and also modify $u(c)$ in Step 13. After such modification and incrementing i, we seek a new $\mathbf{S}_i$. For finding each $\mathbf{S}_i$, we look for the maximum connected component $\mathbf{C}$ of each $\mathbf{R}(c)$, and then set $\mathbf{S}_i$ to be the maximum among all $\mathbf{C}$. In order to reduce the frequency of calculation of connected components, which is a global operation, we make a priority queue D_k of normal cells with $u(c)$ in Step 4. We then repeat dequeue of a normal cell h from D_k to obtain the maximum connected component $\mathbf{C}$ of $\mathbf{R}(h)$ in Step 8. Comparing the size of $\mathbf{C}$ with the maximum among those of other normal cells that are already dequeued from D_k, we finally obtain the currently maximum point set $\mathbf{S}_l$ in Step 9. Note that this loop is repeated until the size of $\mathbf{R}(h)$ is less than s or more than the size of $\mathbf{S}_l$ as described in Step 7. For calculating the maximum connected component of $\mathbf{R}(h)$, we apply a simple method based on a depth-first strategy by using a queue [16]. The time complexity is linear with respect to the size of $\mathbf{R}(h)$.

3.5 Experimental results

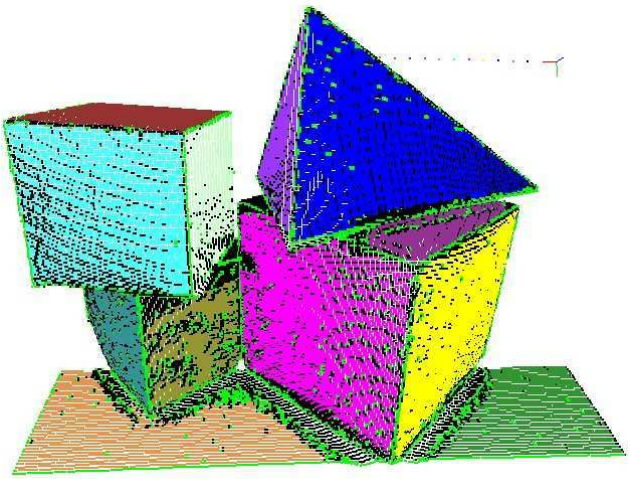
For the experiment, we used six range images of the same blocks, which are taken by a 3D scanner Konica-Minolta VIVID 910 from two different viewpoints with three different resolutions. The range images were transformed into grid-point sets by following the explanation in Subsection 2.4. First, we rejected all non-linear points, as described in Section 2, and then applied Algorithm 1. The results are illustrated in Figs. 7 and 8. In the cases of Fig. 7, the numbers of valid (measured) points are 207459 for (a), 51739 for (b) and 12859 for (c). Among those valid points, we have 184682 locally linear points for (a), 47093 for (b), and 11346 for (c), respectively. Similarly, in the cases of Fig. 8, the numbers of valid (measured) points are 195768 for (d), 48797 for (e) and 12139 for (f). Among those valid points, we have 176697 locally linear points for (d), 44266 for (e), and 10676 for (f), respectively. Tables 2 and 3 show the number of locally linear points that are assigned to each segmented planar surface, and their corresponding color in Figs.7 and 8. We see that 13, 12 and 13 planar surfaces are found in Fig.7 (a), (b) and (c), and 13, 10 and 13 planar surfaces are found in Fig.8 (d), (e) and (f), respectively.

We see in Figs.7 and 8 that non-linear points, colored in light green, appear around edges of block faces, and sometimes appear in faces because of small bumps in faces or noise in the range images. As we set the minimum surface size s, there are locally linear points that construct no planar surface whose size is not less than s around the points, colored in black in the figures. Note that we use 2D connected component labeling in Algorithm 1, instead of 3D connected component labeling, because locally linear points are sparsely distributed in the 3D space, but not in the 2D space.

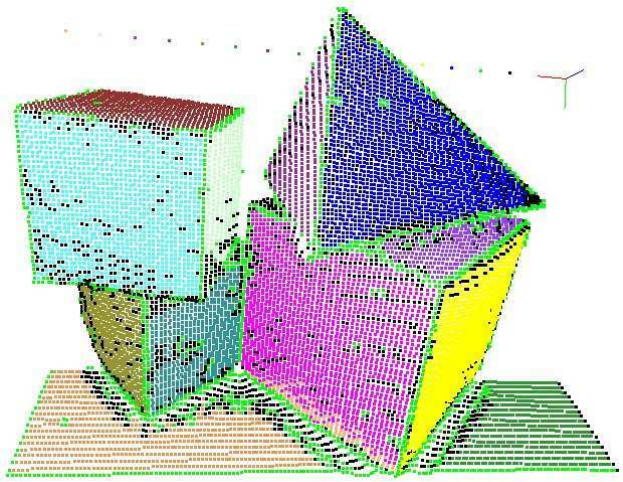
There are physically 12 visible planar surfaces in Fig. 7 and 10 in Fig. 8; there are actually 11 planes in Fig. 7 because a table face is separated into two parts with a right cube. Figures 7 and 8 and Tables 2 and 3 show that all planar surfaces are segmented by our simple algorithm, which require neither complicated parameter setting nor parameter estimation. We should mention that it may bring us rather over-segmentation results when the resolution of an input image is high. For example, the orange and cream points in Fig. 7 (a) (resp. the pale blue and violet points in



(a)

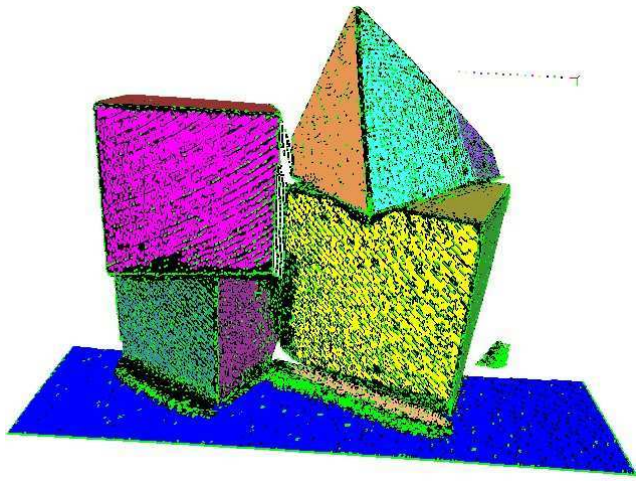


(b)

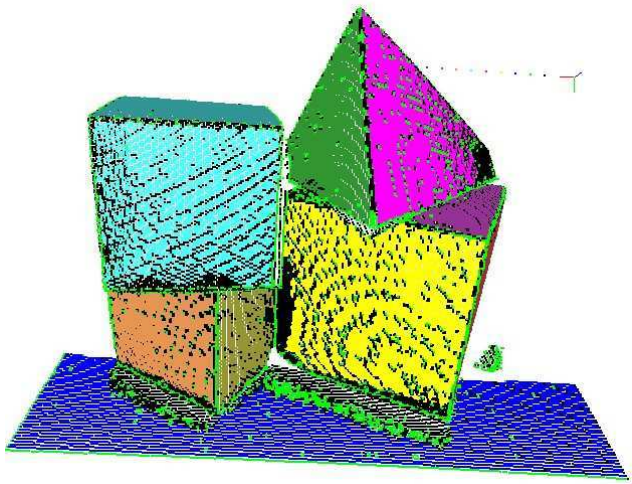


(c)

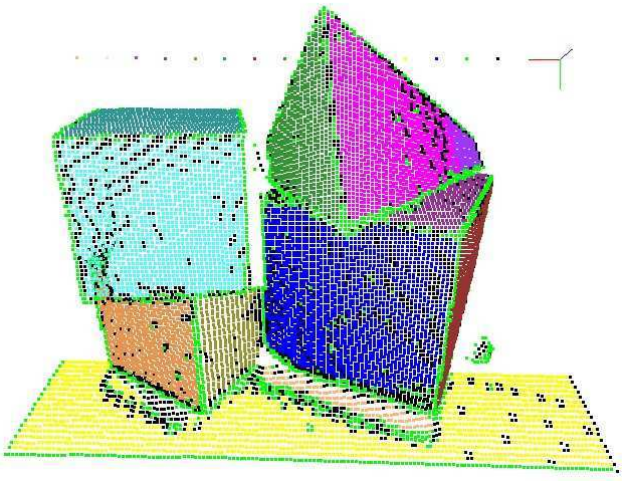
Figure 7 Planar surface segmentation results from range images of blocks, which are taken from the same viewpoint, with different resolutions: the image sizes are 640×480 (a), 320×240 (b), and 640×480 (c). The minimum surfaces sizes s are set to be 1000 (a), 500 (b), and 100 (c), respectively.



(d)



(e)



(f)

Figure 8 Planar surface segmentation results from range images of blocks, which are taken from a different viewpoint from that in Fig. 7, with different resolutions: the image sizes are 640×480 (d), 320×240 (e), 160×120 (f). The minimum surfaces sizes s are set to be 1000 (d), 500 (e), and 100 (f), respectively.

Table 2 Point colors and numbers of segmented planar surfaces in Fig. 7.

color	(a)	(b)	(c)
1 blue	24649	6755	1770
2 yellow	19865	6194	1578
3 pink	17656	5540	1523
4 pale blue	12724	4655	1191
5 orange	12092	3097	699
6 green	10246	2512	573
7 brown	8974	2253	545
8 turquoise	4734	1629	536
9 olive	3787	1517	440
10 purple	3567	985	248
11 violet	3484	979	232
12 moss green	1948	937	223
13 cream	1734		101

Table 3 Point colors and numbers of segmented planar surfaces in Fig. 8.

color	(d)	(e)	(f)
1 blue	28053	6814	1850
2 yellow	21931	6738	1583
3 pink	15074	4315	1182
4 pale blue	13414	4245	1161
5 orange	9981	2994	787
6 green	9153	2547	604
7 brown	8905	2409	589
8 turquoise	8525	2222	542
9 olive	3689	1139	281
10 purple	3470	943	218
11 violet	1852		119
12 moss green	1386		114
13 cream	1145		112

Fig. 8 (d)) should be considered to be in the same region, even if they are separately segmented. We also see that our method is less sensitive to image noise in lower image resolutions; for example, in Fig. 7 (a), there do not exist many linear points on the left cubic face colored in moss green, while more olive and turquoise points are found in Figs. 7 (b) and (c).

As we discussed in Subsection 2.5, we can use larger-size LGPs for the planar surface segmentation. If we use lager-size LGPs, then we will have more normal cells on the discrete Gaussian sphere [17]. This means that each normal cell becomes relatively small so that we can see smaller differences between normal vectors for their distinction. However, we may have a risk of obtaining over-segmentation results. Furthermore, as mentioned before, there are other problems such as obtaining less locally linear points because larger LGPs are more sensitive to noise, and finding a good data structure.

4 Estimation of discrete plane parameters

4.1 Formulation

From each segmented planar-surface set $\mathbf{S}_i$, we estimate its discrete-plane parameters. In this paper, we treat the problem as an linear programming problem. The similar method for the recognition of blurred discrete plane patches can be found in [19].

In order to simplify our problem, we first consider the case that $\omega = |\gamma|$. From (1), we obtain a linear inequality set such that, for all $(x, y, z) \in \mathbf{S}_i$,

$$0 \leq \alpha' x + \beta' y + z + \delta' \leq \epsilon \tag{10}$$

where $\alpha' = \frac{\alpha}{\omega}$, $\beta' = \frac{\beta}{\omega}$, $\delta' = \frac{\delta}{\omega}$. Note that we derive the constraints

$$-1 \leq \alpha' \leq 1,$$
$$-1 \leq \beta' \leq 1$$

from these substitutes. We have another constraint

$$\epsilon \geq 0;$$

if $\epsilon < 1$, the above inequalities are the same as (1). A solution set $(\alpha', \beta', \delta')$ is then obtained by minimizing ϵ under the above constraints. In this framework, if we find a minimum where $\epsilon < 1$, $\mathbf{S}_i$ is recognized as a discrete plane patch exactly; otherwise, $\mathbf{S}_i$ is recognized as a set of grid-points between two parallel planes whose distance is wider than the thickness of a discrete plane. Geometrically, our method looks for two parallel planes such that the z-axial distance between them becomes minimum.

For all the other cases such that $\omega = |\beta|, |\alpha|$, we simply need to modify (10), so that the following inequalities are obtained respectively

$$0 \leq \alpha' x + y + \gamma' z + \delta' \leq \epsilon,$$
$$0 \leq x + \beta' y + \gamma' z + \delta' \leq \epsilon$$

where $\gamma' = \frac{\gamma}{\omega}$. From this substitute, we also derive

$$-1 \leq \gamma' \leq 1.$$

Practically, we simultaneously use the above 3 types of inequality sets to find a parameter set by minimizing ϵ.

4.2 Experimental results

We used a free linear programming solver, lp_solve [24], for our experiments. Tables 4 and 5 show the estimation results for segmented planar surfaces obtained in the previous section, as illustrated in Figs. 7 and 8. Note that we set $\omega = 1$, so that we have $\alpha = \alpha'$, $\beta = \beta'$, and $\gamma = \gamma'$.

We first see that the parameter values of α, β and γ that are obtained for the corresponding planar surfaces, segmented from the range images with different resolutions, are very similar. For example, the first (resp. second) planes in Table 4 (a), (b) and (c), colored in blue (resp. yellow) in Fig. 4, have similar values of α, β and γ.

Table 4 Parameter estimation results of segmented planar surfaces in Fig. 7

(a)

	ϵ	α	β	γ	δ
1	7.63025	-0.490022	-0.0829111	1	1860.94
2	8.57463	1	-0.450926	-0.801559	-1354.71
3	6.06245	0.149226	0.410093	1	1843.06
4	3.98103	0.0159472	1	-0.565851	-1316.86
5	8.3099	1	0.408854	0.9880208	-508.393
6	2.33824	-0.00719424	1	-0.561265	-1310.44
7	2.81801	0.0333703	1	-0.387603	-622.829
8	4.49286	1	0.357483	0.70034	1114.11
9	2.17708	1	-0.10359	0.224703	369.444
10	3.31523	1	-0.086893	-0.182444	-477.839
11	7.37345	0.0296537	1	-0.496269	-927.824
12	3.11816	1	-0.482456	-0.998452	-2065.32
13	2.79155	0.991549	0.388732	1	1860.43

(b)

	ϵ	α	β	γ	δ
1	3.892	-0.492524	-0.086179	1	929.503
2	5.03589	1	-0.4446571	-0.811005	-685.885
3	5.34254	1	0.401473	0.889503	829.663
4	3.32006	0.150978	0.412382	1	921.322
5	2.53489	0.0164474	1	-0.565789	-657.193
6	1.30514	-0.00773908	1	-0.559978	-653.21
7	1.51124	0.0353933	1	-0.387453	-310.698
8	2.53411	1	0.352827	0.699805	557.324
9	1.79852	-0.992593	0.471111	1	1035.22
10	3.67901	0.0308642	1	-0.5	-462.827
11	1.30846	1	-0.0997783	0.228121	188.7
12	1.84748	1	-0.0942873	-0.1797	-235.85

(c)

	ϵ	α	β	γ	δ
1	2.17386	-0.491985	-0.0850801	1	465.34
2	2.60494	1	-0.444444	-0.802469	-339.074
3	3.03361	1	0.403361	0.886555	414.945
4	1.9854	0.153285	0.416058	1	461.65
5	1.50204	0.00816327	1	-0.595918	-343.045
6	0.763006	-0.0115607	1	-0.560694	-327.104
7	0.839786	0.307076	1	-0.387183	-154.698
8	1.22973	1	-0.486486	-0.986486	-510.216
9	1.39538	1	0.352798	0.701946	280.937
10	0.639312	1	-0.103905	0.224355	93.0285
11	1.65236	0.0729614	1	-0.592275	-273.73
12	0.75812	1	-0.0811966	-0.184615	-119.833
13	1.05861	1	0.556777	0.798535	363.923

Table 5 Parameter estimation results of segmented planar surfaces in Fig. 8

(d)

	ϵ	α	β	γ	δ
1	4.77218	0.00593316	1	-0.572602	-1324.12
2	7.05204	0.508394	0.464186	1	1941.96
3	5.69396	-0.222336	0.398329	1	1880.98
4	6.50482	-0.986742	-0.221419	1	1825.7
5	3.56018	1	0.0444174	0.633088	1216.06
6	3.77427	1	-0.211773	-0.319397	-441.095
7	3.36252	0.0244554	1mm1	-0.389848	-622.181
8	6.14341	0.630906	0.492496	1	1781.11
9	7.38064	0.0762753	1mm1	-0.487918	-896.895
10	2.71832	1	-0.225553	-0.466272	-979.645
11	2.72636	1	0.276569	-0.882008	-1577.06
12	2.79741	1	0.0718447	0.083657	-67.6895
13	2.91297	0.115591	1mm1	-0.291331	-758.164

(e)

	ϵ	α	β	γ	δ
1	2.52068	0.00593786	1	-0.573693	-662.817
2	3.75621	0.503386	0.465011	1	971.786
3	4.43575	1	0.229012	-0.990006	-898.568
4	3.29534	-0.216321	0.409326	1	940.894
5	2.88372	0.613953	0.47907	1	893.614
6	1.99154	1	0.0444047	0.62931	605.345
7	1.9594	1	-0.212029	-0.321437	-222.269
8	1.63415	0.0243902	1	-0.390244	-311.049
9	1.46143	1	-0.226508	-0.464236	-487.532
10	3.10073	0.0680581	1	-0.493648	-454.253

(f)

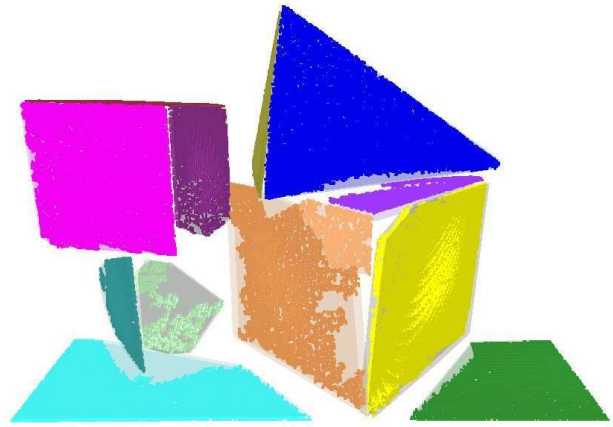
	ϵ	α	β	γ	δ
1	2.45884	0.51417	0.460189	1	487.51
2	2.20412	0.013526	1	-0.57055	-328.931
3	2.59351	1	0.228652	-0.997177	-451.424
4	1.99307	-0.228571	0.393939	1	471.536
5	1.82812	0.617188	0.476562	1	447.953
6	1.18626	1	0.0463576	0.631623	304.741
7	0.935347	1	-0.21142	-0.316659	-108.528
8	0.938095	0.0238095	1	-0.390476	-155.233
9	0.746032	1	-0.222222	-0.460317	-241.714
10	1.2268	0.0515464	1	-0.474227	-217.897
11	1.04615	1	0.282051	-0.866667	-385.815
12	0.545455	1	0.0606061	0.0909091	-12.8788
13	0.831683	0.108911	1	-0.316832	-201.564

Concerning to the parameter δ, the values in Tables 4 (a) and 5 (d) (resp. Tables 4 (b) and 5 (e)) are almost four times (resp. twice) as large as those in Tables 4 (c) and 5 (f), respectively. The reason is that the grid space of Figs. 7 (a) and 8 (d) (resp. Figs. 7 (b) and 8 (e)) is four times (resp. twice) as large as that of Figs. 7 (c) and 8 (f), because of their image resolutions. Note that we set the grid interval to be 1 for the parameter estimation.

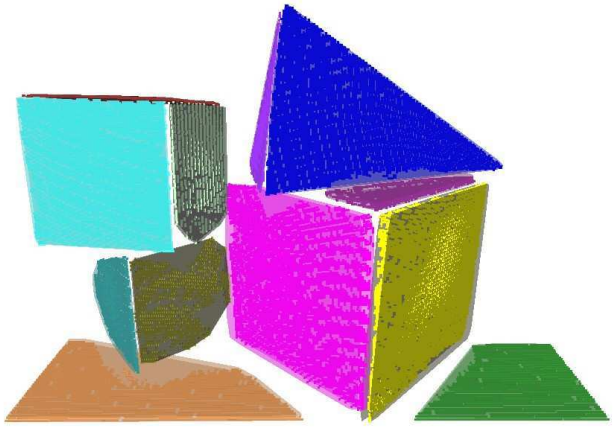
From Tables 4 and 5, we also see that it is rare that ϵ becomes less than 1, specially when the image resolution is high. In other words, our segmented planar surfaces can be exactly discrete planes, when the resolution becomes lower. The tables show that the higher the image resolution, the larger the value ϵ. Since each segmented planar surface contains many grid points when the image resolution is high, as seen in Table 2, it can generate a thicker discrete plane. Figure 9 illustrates the estimated discrete plane with a minimum thickness, namely, the two parallel planes with a minimum distance, for each segmented point set in Fig. 7. In Fig. 9, we see that there is no isolated point in any segmented point set, thanks to the non-linear point rejection and the connected component labeling in Algorithm 1. Therefore, the thickness may be related to the surface curvedness of a segmented point set, as well as the shape and the size. It might be interesting to study how we can reduce the thickness ϵ by changing the image resolution, for the aim of inventing a multiscale method for range image registration by using planar surfaces, for example.

5 Conclusion

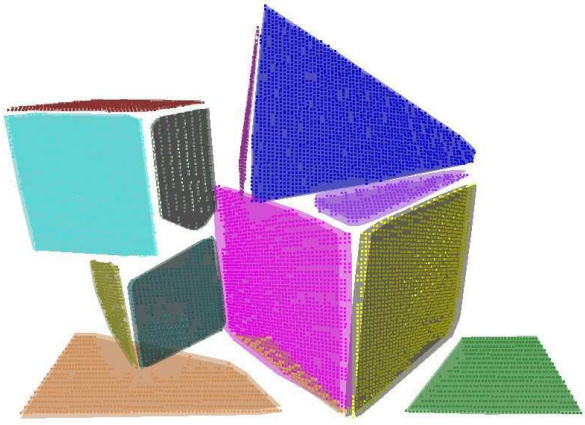
In this paper, we present a discrete version of the hybrid method for planar surface segmentation from a 3D grid-point set. Our method simply requires two types of look-up tables, such as the binary LGP table (linear or non-linear) and the normal cell list with respect to each linear LGP, and does not require any parameter setting/estimation. The experimental results in Figs. 7 and 8 show us that our method is useful for planar surface segmentation from a point cloud, because it takes into account not only quantization errors but also noise. We also present a method for estimating discrete-plane parameters, which is also based on discrete geometry. Theoretically, exact discrete planes must be obtained if input is an ideal image, i.e., it does not contain noise, but contains only quantization errors. However, our estimation results in Tables 4 and 5 show us that exact discrete planes are rarely obtained for practical images. This is because input images contain noise as well as quantization errors. Therefore, we need to eliminate such noise, for example, by reducing image resolutions, before applying our method based on discrete geometry. As our method is fully discrete and such discreteness may help us to build up a multiscale approach, we will reorient our future work to inventing a multiscale method for range image registration by using discrete planes, for example. We expect that our approach will provide a rough registration result with less computation.



(a)



(b)



(c)

Figure 9 Estimated two parallel planes with a minimum thickness for each segmented point set in Fig. 7.

References

[1] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Second Edition, Cambridge University Press, Cambridge, 2003.

[2] H.-Y. Shum, K. Ikeuchi, R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp.854–867, 1995.

[3] G. Papaioannou, E.-A. Karabassi, T. Theoharis. Segmentation and surface characterization of arbitrary 3D meshes for object reconstruction and recognition. in *Proceedings of Ineternational Conference on Pattern Recognition*, IEEE, pp.734–737, 2000.

[4] G. Taylor, L. Kleeman. Robust range data segmentation using geometric primitives for robotics applications. in *Proceedings of the IASTED International Conference on Signal and Image Processing*, pp.467–472, 2003.

[5] P. J. Besl, R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp.167–192, 1988.

[6] O. D. Faugeras, M. Hebert, E. Pauchon. Segmentation of range data into planar and quadric patches. In *Proceedings of Computer Vision and Pattern Recognition*, pp.8–13, 1983.

[7] D. Cohen-Steiner, P. Alliez, M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, vol. 23, no. 3, pp.905–914, 2004.

[8] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp.673–689, 1996.

[9] D. Marshall, G. Lukacs, R. Martin. Robust segmentation of primitives from range data in the presence of gemetric degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp.304–314, 2001.

[10] H. Chen, B. Bhanu. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, vol. 28, pp.1252–1262, 2007.

[11] K. Haris, S. N. Efstratiadis, N. Maglaveras, A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, vol. 7, no. 12, pp.1684–1699, 1998.

[12] K. Köster, M. Spann. MIR: an approach to robust clustering - application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp.430–443, 2000.

[13] D. Zhao, X. Zhang. Range-data-based object surface segmentation via edges and critical points. *IEEE Transactions on Image Processing*, vol. 6, no. 6, pp.826–830, 1997.

[14] I. Stamos, P. K. Allen. 3D model construction using range and image data. In *Proceedings of Computer Vision and Pattern Recognition*, vol. 1, pp.531–536, 2003.

[15] N. Yokoya, M. D. Levine. Range image segmentation based on differential geometry: a hybrid approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp.643–649, 1989.

[16] R. Klette, A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kauffmann, San Francisco, 2004.

[17] Y. Kenmochi, L. Buzer, A. Sugimoto, I. Shimizu. Digital planar surface detection using local geometric patterns. submitted to *Processings of Discrete Geometry for Computer Imagery*.

[18] D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, J.-M. Chassery. On digital plane preimage structure. *Discrete Applied Mathematics*, vol. 151 no. 1-3, pp.78–92, 2005.

[19] L. Provot, L. Buzer, I. Debled-Rennesson. Recognition of blurred pieces of discrete planes. In *Processings of Discrete Geometry for Computer Imagery*, Springer-Verlag, LNCS 4245, pp.65–76, 2006.

[20] J.-P. Reveillès. Combinatorial pieces in digital lines and planes. in *Vision Geometry IV*, SPIE, vol. 2573. pp.23–34, 1995.

[21] Y. Kenmochi, A. Imiya. Combinatorial boundary of a 3D lattice point set. *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp.738–766, 2006.

[22] G. M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, New York, 1998.

[23] B. K. P.Horn. Extended Gaussian images. *Proceedings of the IEEE*, vol. 72, no. 12, pp.1671–1686, 1984.

[24] lp_solve. http://lpsolve.sourceforge.net/5.5/