Planar surface segmentation of a point cloud using local geometric patterns

Yukiko Kenmochi, Lilian Buzer, Akihiro Sugimoto, Ikuko Shimizu

Planar surface segmentation of a point cloud using local geometric patterns

Yukiko Kenmochi[1] Lilian Buzer[1] Akihiro Sugimoto[1,2] Ikuko Shimizu[3]
[1]Institut Gaspard-Monge, Université Paris-Est, France
[2]National Institute of Informatics, Japan
[3]Tokyo University of Agriculture and Technology, 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 planar-surface-point set, we also estimate parameters of a discrete plane by minimizing its thickness.

1. Introduction

This paper presents a method for segmenting a 3D point cloud into planar surfaces using recently obtained discrete-geometry results. Conventional approaches are classified into three categories: region-based approach, edge-based approach and hybrid approach. The first one merges points that have similar region properties calculated from their neighboring points such as normal vectors and curvatures [1]. Because calculated properties are sensitive to noise and quantization errors, it is known that they cause over segmentation. In the second approach, edges are searched for such that they separate regions by using depth discontinuities [9]. Because edges are not always extracted as connected curves, they cause under segmentation, contrary to the first ones. The third approach is hybrid between the two approaches, so that they import both their merits [7, 8]. For a special case in the third approach, a planar segmentation method is proposed based on locally planar points, considering points that are not locally planar to be potentially edge points [7].

In this paper, we present a discrete version of the hybrid method. 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 [5]. Contrarily to the continuous case, there exist a finite number of local geometric patterns (LGPs) appearing on discrete planes, called linear LGP [3]. In fact, points that have linear LGP can be considered to be discrete version of locally planar points [7]. In addition, such a linear LGP does not possess the unique normal vector but a set of normal vectors [2]. 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 LGP, 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 such advantages by applying our algorithm to 3D point clouds such as range images. In order to evaluate our segmentation results, we also present a method for estimating discrete plane parameters from each segmented planar surface by minimizing its thickness. Because the thickness indicates the segmentation inaccuracy, namely the curvedness of a segmented planar surface, 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 [5].

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

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

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 an object point.

Among those different LGPs, we investigated which LGP can appear on discrete planes [3]. 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

$$\begin{aligned}\mathbf{F} &= \mathbf{D}(\mathbf{P}) \cap \mathbf{Q}(\boldsymbol{x}) \\ &= \{(p,q,r) \in \mathbf{Q}(\boldsymbol{x}) : 0 \leq \alpha p + \beta q + \gamma r + \delta < \omega\} \quad (2)\end{aligned}$$

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 LGP, satisfying (2). Such LGP are called linear LGP. Since this problem is considered to be the feasibility of the inequalities of (2) 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, [3] 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 [5], 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

$$0 \leq \alpha \leq \beta \leq 1,\ \gamma = 1. \quad (3)$$

In order to visualize the shapes of linear LGPs in Fig. 1, we add polyhedral meshes generated by applying a discrete-marching-cube-like method for the 18-neighborhood system [4] to a digitized half space. Interior points of objects 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 is exactly a point that 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.

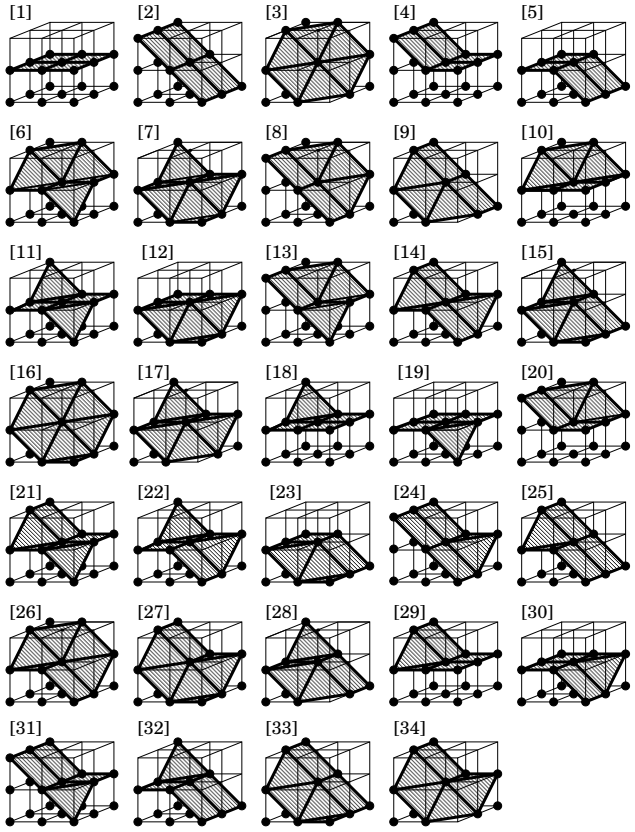


Figure 1. The 34 linear LGPs.

2.4. Non-linear point rejection

By simply checking the LGP linearity, we can therefore reject non-linear points from a grid-point set, since 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). In actual experiments, we see that isolated points that are considered to be noise are automatically rejected as well as points around surface edges. However, it is also observed that some points around an edge are not rejected, especially when two adjacent surfaces joining the edge intersect at an obtuse angle. This fact implies that a simple post-processing, such as the connected component labeling [5] of a non-rejected point set, does not always give

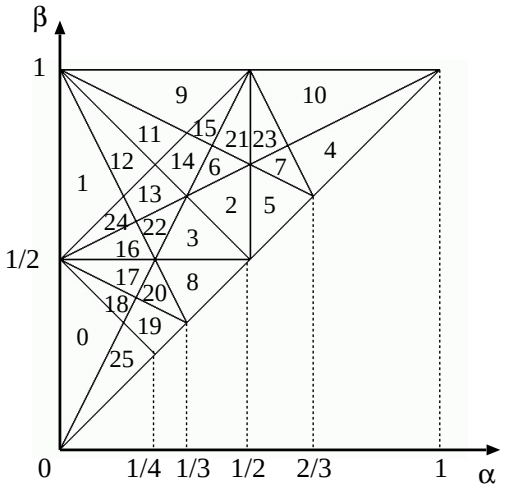


Figure 2. Normal cells on the $\alpha\beta$-plane with constraint (3).

satisfactory results for planar surface segmentation.

3. Planar surface segmentation of locally linear points

In order to solve the above 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 [2]. 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 (1) for all points of $\mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$. It means that the preimage is given by a convex polytope in the parameter space [2].

Because 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 to the inequality set of (1) 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 [3].

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

The results are derived in the space (α, β) from linear LGP with the constraints (3). 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 (3). Figure 2 illustrates that the inequality set of (1) for all $(p, q, r) \in \mathbf{D}_{\mathbf{Q}(\boldsymbol{x})}(\mathbf{P})$ divides the triangular region in the space (α, β) into triangular or quadrilateral polygons, 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 such a set of normal cells whose union corresponds to a convex polygon representing a set of feasible normal vectors for each linear LGP depicted in Fig. 1.

3.2 Discrete Gaussian sphere

The 26 normal cells in Fig. 2 are generated with the constraints (3). We embed these normal cells into the 3D space (α, β, γ) with $\gamma = 1$, as illustrated in Fig. 3 (left). The triangle surrounded by thick lines in Fig. 3 (left) corresponds to the triangular region that is the union of normal cells in Fig. 2. Once the normal cells are embedded into the space (α, β, γ), we make the congruous ones by applying to them

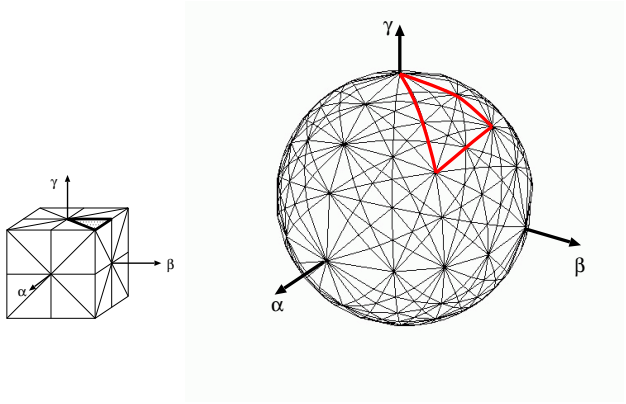


Figure 3. The cubical Gaussian sphere and the discrete Gaussian sphere.

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. 3 (left), 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. 3 (right). The unit sphere separated by normal cells is called 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. 3 (right) corresponds to the triangle surrounded by thick lines in Fig. 3 (left) that corresponds to the union of normal cells in Fig. 2. 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 a 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 [6], 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 $\mathbf{Z}^3$. 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

$$\mathbf{R}(c) = \{\boldsymbol{x} \in \mathbf{V} : c \in \mathbf{I}(\boldsymbol{x})\}. \quad (4)$$

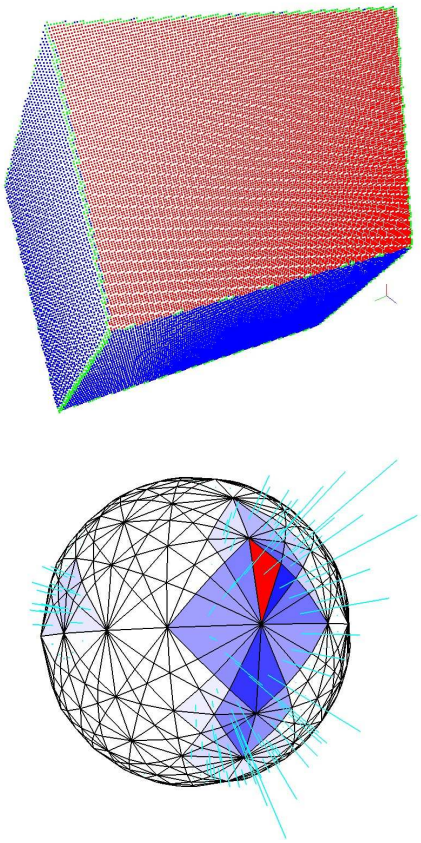


Figure 4. A synthetic 3D image of a box (top) and its unified discrete Gaussian image (bottom).

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

$$u(c) = |\mathbf{R}(c)|. \quad (5)$$

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

The concept of unified discrete Gaussian images is similar to that of extended Gaussian images [6]. The differences from extended Gaussian images are the followings: the function (5) 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 (5) 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 4 shows an example of the unified discrete Gaussian images for a digitized box. Concerning cell colors on the discrete Gaussian sphere in Fig. 4 (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. 4 (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. 4 (bottom). Figure 4 shows that we can extract a set of grid points that belong to a digital plane $\mathbf{D}(\mathbf{P})$ by choosing a "right" normal cell, for example, a red one. This is based on the following fact; 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 $R(c)$, 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 $R(c)$. Because 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 $R(c)$, 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 [5]. The time complexity is linear with respect to the size of $\mathbf{R}(h)$.

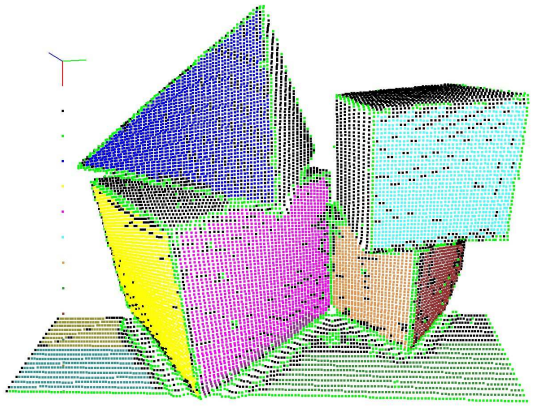
Algorithm 1: Planar surface segmentation

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

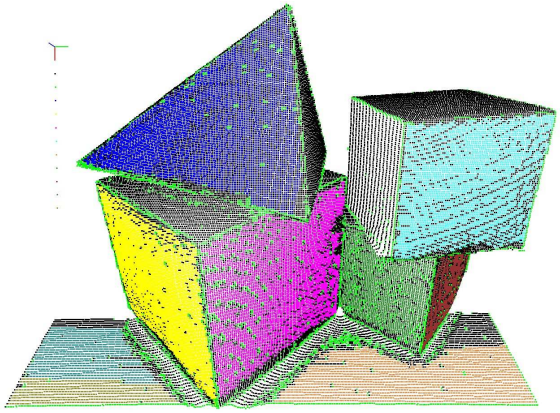
```
1  begin
2      initialize a label such that l = 0;
3      repeat
4          make a queue D_k of normal cells with
           priorities of values u(c);
5          increment l and initialize S_l = ∅;
6          set h to be the highest priority cell in D_k and
           remove it from D_k;
7          while |R(h)| > max(s − 1, |S_l|) do
8              set C to be the maximum connected
               component of R(h);
9              if |C| > |S_l| then set S_l = C;
10             reset h to be the highest priority normal
               cell in D_k and remove it from D_k;
11         if |S_l| ≥ s then
12             forall c such that u(c) ≠ 0 and
               R(c) ∩ S_l ≠ ∅ do
13                 reset R(c) = R(c) \ S_l and
                   u(c) = |R(c)|;
14     until |S_l| < s;
15     return S_i for i = 1, 2, ..., l − 1;
16 end
```

3.5. Experimental results

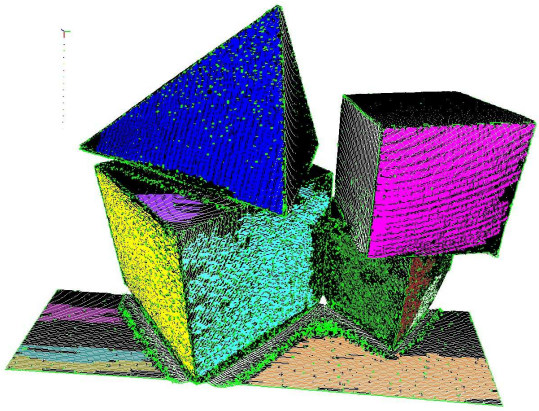
We show results of planar surface segmentation from six range images of the same blocks, which are taken from two different viewpoints with three different resolutions. The results are illustrated in Figs. 5 and 6. In the cases of Fig. 5, the numbers of valid (measured) points are 12858 for (a), 51740 for (b) and 207448 for (c). Among those valid points, we have 11343 locally linear points for (a), 47034 for (b), and 185566 for (c), respectively. Similarly, in the cases of Fig. 6, the numbers of valid (measured) points are 12142 for (d), 48802 for (e) and 195765 for (f). Among those valid points, we have 10663 locally linear points for (d), 43981 for (e), and 176619 for (f), respectively. Table 2 show the number of locally linear points that are assigned to each segmented planar surface, and their corresponding color in Figs.5 and 6. We see that 10, 9 and 12 planar surfaces are found in Fig.5 (a), (b) and (c), and 8, 8 and 11 planar surfaces are found in Fig.6 (d), (e) and (f), respectively.



(a)

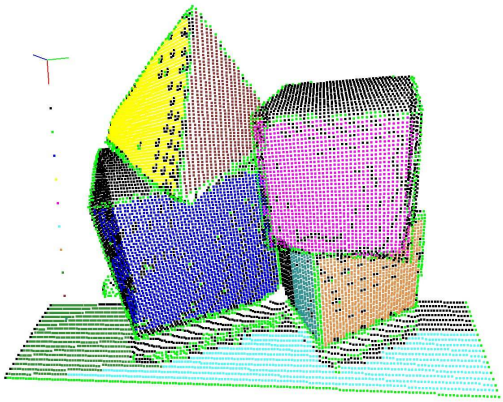


(b)

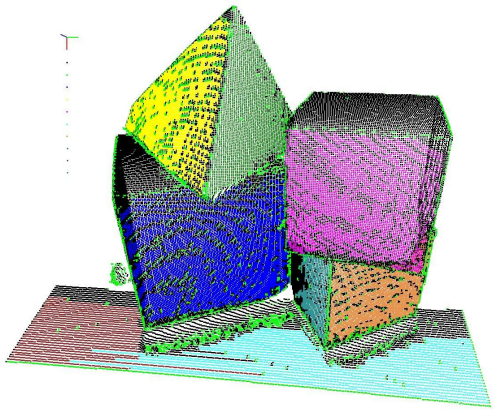


(c)

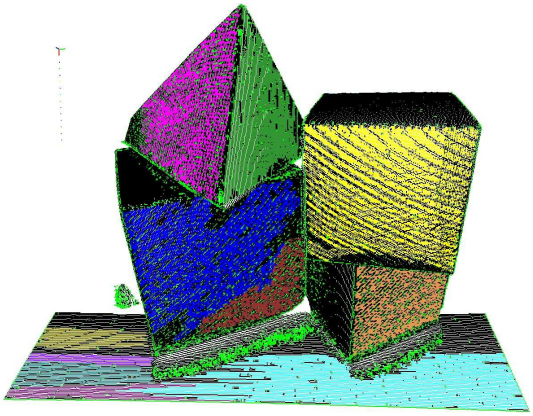
Figure 5. Planar surface segmentation results from range images of blocks, which are taken from the same viewpoint, with different resolutions: the image sizes are 160×120 (a), 320×240 (b), 640×480 (c). The minimum surfaces sizes s are set to be 100 (a), 500 (b), and 1000 (c), respectively.



(d)



(e)



(f)

Figure 6. Planar surface segmentation results from range images of blocks, which are taken from a different viewpoint from that in Fig. 5, with different resolutions: the image sizes are 160×120 (d), 320×240 (e), 640×480 (f). The minimum surfaces sizes s are set to be 100 (d), 500 (e), and 1000 (f), respectively.

Table 2. Colors and point numbers of segmented planar surfaces in Figs. 5 and 6.

color	(a)	(b)	(c)	(d)	(e)	(f)
1 blue	1750	6529	23393	1914	6304	15823
2 yellow	1557	6125	19822	1280	4254	13809
3 pink	1536	5573	16645	1171	4006	12879
4 pale blue	1180	4408	14377	822	3505	12198
5 orange	595	2627	9375	800	2929	9098
6 green	579	1901	3022	638	2437	8867
7 brown	418	1620	2376	612	2321	5009
8 turquoise	361	1424	1609	220	844	2266
9 olive	197	799	1419			2169
10 purple	112		1418			1857
11 violet			1160			1212
12 moss green			1036			

We see in Figs.5 and 6 that non-linear points, colored light green, appear around edges of block faces, and sometimes appear in faces because of noise in the range images. Because 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 black in the figures. Locally linear points become black, if they are obtained too sparsely in the 3D space to be connected, so that they are seen as some of block faces in Figs.5 and 6. Otherwise, they are considered to have bumps on the whole, even though they are locally linear. From Figs. 5 and 6, we can conclude that most of the block surfaces are segmented by our simple algorithm, which require neither complicated parameter setting nor parameter estimation.

4. Estimation of discrete plane parameters

4.1 Formulation

From each segmented planar-surface set $\mathbf{S}_i$, we estimate its discrete-plane parameters as follows. 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 $(x, y, z) \in \mathbf{S}_i$,

$$0 \leq \alpha' x + \beta' y + z + \delta' \leq \epsilon \tag{6}$$

where $\alpha' = \frac{\alpha}{\omega}$, $\beta' = \frac{\beta}{\omega}$, $\delta' = \frac{\delta}{\omega}$, $\epsilon \geq 0$. A solution set $(\alpha', \beta', \delta')$ is then obtained by minimizing ϵ under the above constraints. In this framework, if $\epsilon < 1$, $\mathbf{S}_i$ 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 distance between them becomes minimum.

For all the other cases such that $\omega = -\gamma, \beta, -\beta, \alpha, -\alpha$, we simply need to modify (6), so that the following inequalities are obtained respectively

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

Table 3. Parameter estimation results of segmented planar surfaces in Fig. 5

(a)

	ϵ	α	β	γ	δ
1	2.30846	0.0895522	0.512438	1	-480.756
2	2.51369	0.444688	1	0.779847	345.085
3	3.47581	-0.475806	-1.16935	1	-480.306
4	2.01022	-0.434227	-0.157088	1	-477.374
5	1.26786	-0.482143	1	-0.946429	-508.393
6	1.47327	1	0.0178218	-0.558416	-335.198
7	1.125	0.351563	1	0.664062	274.57
8	0.605714	1	-0.0114286	-0.554286	-334.337
9	0.5	1	0	-0.5	-305.5
10	1.51563	-0.78125	-1.32812	1	-468.875

(b)

	ϵ	α	β	γ	δ
1	3.90025	0.0910617	0.51279	1	-962.483
2	4.89735	0.4446018	1	0.780531	691.458
3	6.31767	-0.474464	-1.16547	1	-963.488
4	3.24647	-0.433281	-0.153846	1	-954.964
5	1.54492	1	0.0212766	-0.554374	-668.843
6	1.72454	-0.478588	1	-0.956019	-1027.81
7	2.23761	0.356076	1	0.671419	555.339
8	0.791798	1	-0.0126183	-0.536278	-651.688
9	0.717703	1	-0.00956938	-0.564593	-680.029

(c)

	ϵ	α	β	γ	δ
1	7.38188	0.086763	0.507398	1	-1926.6
2	9.66667	0.444444	1	0.777778	1378.33
3	6.23908	-0.422621	-0.149001	1	-1910.67
4	10.8503	-0.487362	-1.15165	1	-1924.55
5	2.1997	1	0.0222728	-0.558241	-1347.4
6	2.44488	-0.479921	1	-0.955906	-2056.17
7	2.26689	0.375	1	0.726351	1217.36
8	0.886656	1	-0.0103611	-0.562951	-1358.17
9	0.875	1	-0.00961538	-0.572115	-1376.38
10	0.808511	1	-0.0141844	-0.529255	-1290.44
11	4.24536	1.81477	0.0910698	1	-1930.37
12	2.17989	0.39418	1	0.626984	1009.64

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

where $\gamma' = \frac{\gamma}{\omega}$. Practically, we simultaneously use the above 6 types of inequalities to find a parameter set minimizing ϵ.

4.2 Experimental results

Tables 3 and 4 show the estimation results for segmented planar surfaces obtained in the previous section, as illustrated in Figs. 5 and 6. We 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. Concerning to the parameter δ, the values in Table 3 (b) and Table 4 (e) (resp. Table 3 (c) and Table 4 (f)) are almost twice (resp. four times) as large as those in Table 3 (a) and Table 4 (c). The reason is that the grid space of Fig. 5 (b) and Fig. 6 (e) (resp. Fig. 5 (c) and Fig. 6 (f)) is twice (resp. four times) as large as that of

Table 4. Parameter estimation results of segmented planar surfaces in Fig. 6

(d)

	ϵ	α	β	γ	δ
1	2.53866	-0.481622	-0.539924	1	-505.439
2	2.78801	-0.235546	1	0.940043	446.441
3	2.02985	-0.41791	0.238806	1	-489.358
4	1.02676	1	0.0126761	-0.550235	-331.395
5	1.92059	-0.505882	-0.638235	1	-464.826
6	0.760252	1	-0.0115668	-0.539432	-326.766
7	1.74775	-0.0720721	-1.63964	1	-500.802
8	0.660584	-0.0720721	1	-0.448905	-245.204

(e)

	ϵ	α	β	γ	δ
1	4.0725	-0.486857	-0.541143	1	-1010.1
2	5.03725	-0.232787	1	0.936131	890.082
3	3.22654	-0.417907	0.232147	1	-978.429
4	1.75286	1	0.0111605	-0.549891	-663.497
5	3.50924	-0.50308	-0.652977	1	-927.704
6	3.09351	-0.073294	-1.64027	1	-1000.16
7	1.07945	1	-0.00797011	-0.546077	-660.958
8	1.20741	-0.237037	1	-0.451852	-493.904

(f)

	ϵ	α	β	γ	δ
1	4.88192	-0.462209	-0.519922	1	-2020.35
2	6.38296	-0.413744	0.231031	1	-1956.85
3	8.11213	-0.227372	1	0.948015	1804.6
4	2.76525	1	0.0120939	-0.553442	-1335.69
5	6.59621	-0.51084	-0.651762	1	-1854.09
6	5.85422	-0.0745445	-1.64605	1	-1998.81
7	4.33574	-0.520009	-0.586203	1	-2016.94
8	0.981103	1	-0.0106891	-0.559076	-1350.38
9	0.888889	1	-0.0121382	-0.531279	-1294.01
10	1.06774	1	-0.00811321	-0.55717	-1346.02
11	0.921053	1	-0.0115132	-0.541118	-1314.11

Fig. 5 (a) and Fig. 6 (d), because of their image resolutions. Note that we always set the grid interval to be 1, when we make a grid space from a range image.

From Tables 3 and 4, we also see that it is rare that ϵ becomes less than 1. In other words, most of our segmented planar surfaces cannot be exactly discrete planes. Moreover, 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 very thicker discrete plane. It is possible that the thickness is related to the spread of grid points in a segmented planar surface as well as the point number. It might be interesting to study how we can reduce the thickness ϵ by changing the image resolution, with 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. 5 and 6 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. Our estimation results in Tables 3 and 4 show us that exact discrete planes are rarely obtained for practical images, because of their noise. 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, practically, it is no wonder that input image contains noise as well as quantization errors. Therefore, we have to derive a conclusion that we need to eliminate such noise, for example, by reducing image resolutions, before applying our method based on discrete geometry. Because our method is fully discrete and such discreteness helps us to build up a multiscale approach, we will reorient our future work to inventing the 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.

References

[1] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, 1988.

[2] D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, and J.-M. Chassery. On digital plane preimage structure. *Discrete Applied Mathematics*, 151(1-3):78–92, 2005.

[3] Y. Kenmochi, L. Buzer, A. Sugimoto, and I. Shimizu. Digital planar surface detection using local geometric patterns. *submitted to Processings of Discrete Geometry for Computer Imagery*, 2008.

[4] Y. Kenmochi and A. Imiya. Combinatorial boundary of a 3d lattice point set. *Journal of Visual Communication and Image Representation*, 17(4):738–766, 2006.

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

[6] B. K. P.Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984.

[7] I. Stamos and P. K. Allen. 3d model construction using range and image data. *Proceedings of Computer Vision and Pattern Recognition*, 1:531–536, 2003.

[8] N. Yokoya and M. D. Levine. Range image segmentation based on differential geometry: a hybrid approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):643–649, 1989.

[9] D. Zhao and X. Zhang. Range-data-based object surface segmentation via edges and critical points. *IEEE Transactions on Image Processing*, 6(6):826–830, 1997.