Polygonal Model Creation with Precise Boundary Edges from a Mobile Mapping Data

***Ryuji Miyazaki*†, *Makoto Yamamoto*†† *Koichi Harada*†††**

†Faculty of Psychological Science, Hiroshima International University[1]
††Sanei Co.[2]
†††Hiroshima University[3]

Summary
In this paper, we propose a method for creating a three-dimensional polygonal model reflecting precise boundary of physical objects from a point cloud which has strong anisotropic distribution of points obtained by a mobile mapping system. Our method creates boundary information of physical objects based on the planar region segmentation by the line-based region growing approach from a point cloud. Then, the boundary information is used for constraining an edge of a polygon in the polygonal model creation procedure. As a result, boundary edges of actual physical objects appear as edges of the polygonal model adequately. The accuracy of boundary edge representation is evaluated by applying the proposed method to a point cloud obtained by simulation of scanning a road polygonal model. Finally, we create a polygonal model from the actual mobile mapping data.
Key words:
Polygonal model, boundary edge, point cloud, anisotropic distribution, mobile mapping system.

1. Introduction

A recent development of three-dimensional laser scanning system make it possible to obtain high density point cloud easily. The mobile mapping system (MMS) which is equipped with such a high density laser scanning system measures an environment around a road precisely by simply traveling on a road at a standard speed [1]. Therefore, the MMS does not need any traffic regulation to measure an environment around a road. This is very helpful to the survey of road environment for the purpose of maintenance and management. An application of a large scale three-dimensional point cloud for a management of the infrastructure facilities like road environments is one of the important topics. Numerous techniques are proposed for practical purposes [2–9].

For further application of a point cloud, the technique of constructing a three dimensional shape model of a structural object is required. The appearance of a measured structural object is able to be displayed by drawing points using their three-dimensional coordinate. However, we are able to offer more information of scene by creating a three dimensional shape model like a polygonal model. The polygonal model makes it possible to express a surface by shading. In terms of scientific visualization, it is well known that the surface with shading based on the information of a normal vector is useful for expressing change of a surface.

In other application, a cross-section view is required for maintenance of a road. The buffer method [10] is one of the methods creating a cross-section view from measured points. This method selects measured points near a cross-section. Then, the selected points are projected onto the plane that a cross-section will be created. Finally, the least squares approximation line is created from the projected points for a cross-section. Belton et al. also project measured points onto the plane and find an approximation of a joint part of a road and a curbstone [11]. However, the precision of the result which is obtained by the least squares approximation is limited. A blank portion of measured points is interpolated by generating polygonal model using measured points. Therefore, we are able to obtain a precise cross-section view easily by calculating intersection of the plane of a cross-section with the polygonal model which approximates a surface of an object appropriately.

A polygonal model needs the geometrical information as the three dimensional coordinate of vertices, and the topological information as the connection of vertices constituting a polygon. There are two factors should be considered for creating a precise polygonal model. One is to select appropriate vertices from a huge number of measured points in order to constitute polygons representing the surface of an object efficiently. The other is to constitute appropriate polygons for interpolating a surface of the object. In the case of creating a polygonal model from a point cloud obtained by a MMS, we need to obtain the topological information for constituting polygons from a point cloud. He et al. proposed the method to create a polygonal model by connecting the nearest point in two consecutive scan lines [12]. This method is able to create a polygonal model very fast.

However, in some cases, it is difficult to create an appropriate polygonal model due to the reason caused by some situations of a point cloud. In particular, inappropriate edges often appear around the boundary of planar surfaces. This decreases the fidelity of a polygonal model and the precision of the derived information like a

Manuscript received November 5, 2016
Manuscript revised November 20, 2016

cross-section view. Furthermore, a polygonal model can be used as the boundary condition for the physical simulation like a flood simulation. Therefore, the boundary edge between structures should be appeared precisely in a polygonal model.

The reason inappropriate boundary edges are created is because the structural information is not included in the process of polygon creation. An appropriate polygonal model can be created by reflecting the structural information of a target object. One of the effective techniques for obtaining the structural information from a point cloud is the planar region extraction. The boundary between different planar regions can be considered as the boundary edge information. Therefore, such a boundary edge should be reflected in the process of polygon creation.

The region growing approach is widely used as the method for planar region extraction from a point cloud. However, in the case of the point cloud which is obtained by a MMS, the traditional point-based region growing approach may not obtain reasonable result due to anisotropic distribution of points.

In this paper, we propose the method to create a polygonal model which is able to precisely represent boundaries of surfaces constituting a physical object. Our approach obtains the information of the boundary of the surfaces from an input point cloud which has anisotropic distribution by using the line-based planar region segmentation. Then, such an information is used as the constraints for representing the boundary edge in the process of creating polygons.

The remaining of this paper consists of following sections. The second section mentions the advantages of the line-based processing which is adopted to our proposal. In the third section, we explain the method for obtaining the boundary edge information from a point cloud. In the fourth section, we create the polygonal model considering the boundary edge information which is obtained by the previous section. The evaluation of the proposed method is conducted in the section five. Finally, we conclude this paper in the section six.

2. Advantages of Line-Based Processing

2.1 Anisotropic Point Distribution of a Point Cloud

Recent laser scanning system is able to measure up to a million points per second. In the direction traversing a road, points are measured very densely. Actually, the interval of measured points is often a few millimeters. However, the interval of measured points in the traveling direction of the MMS depends on the rotating speed of the laser irradiation part and the speed of which the MMS travels. The rotation period of the laser irradiation part is much longer than the period of the laser irradiation. In the traveling direction of the MMS, the interval of measured points is often a few hundreds of millimeters. Hence, the density of point distribution is greatly different depending on the direction (Fig. 1).

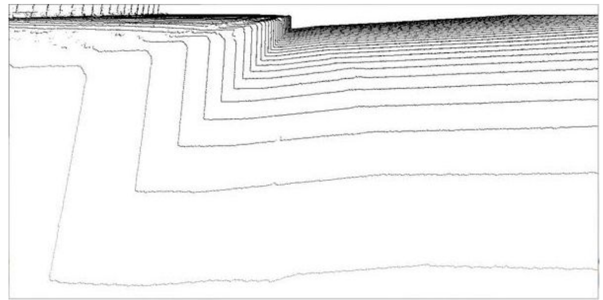

Fig. 1 Anisotropic point distribution of measured points.

This situation causes the problem to the method of calculating geometric information using neighborhood points because it is necessary to define the neighborhood range too wide in the sparse direction (Fig. 2(a)). If the neighborhood range is not wide enough, the neighborhood includes points in one direction only (Fig. 2(b)). In the case of Fig. 2(a), we are not able to estimate a normal vector precisely at the point on the planar region around the

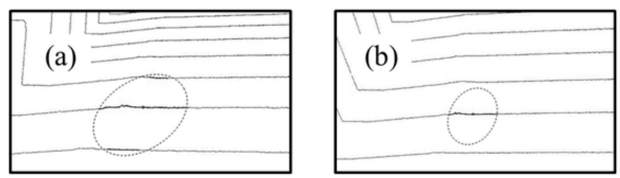

Fig. 2 Two cases of the neighborhood region for calculation of geometric information.

boundary of structures. Fig. 3 shows an example of such a case. In the vicinity of the boundary of a road and a curbstone, the estimated normal vector is tilted because the neighborhood range includes points of the vertical plane even though the target point is on the horizontal plane. The geometrical information such as the normal vector is

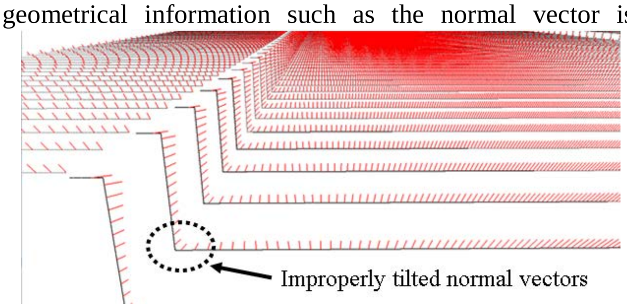

Fig. 3 Inadequate estimation of normal vector around the boundary between road and curbstone.

important factor to classify points. Therefore, it is necessary to calculate such a geometrical information precisely for processing a point cloud.

2.2 Line Segment Creation from MMS Data

First, we separate an input point cloud into sequences of points that represent scanning lines. Each point of MMS data contains the angle information of which the laser has been irradiated. The angle in an identical scanning line increases monotonically from $0°$ to $360°$. That is, the point associated with $0°$ is at the beginning of a scanning line. In practice, there is the case that the $0°$ point is not measured. So, let the beginning of a scanning line be the point where the value of angle decreases from the previous point.

Next, we apply the Douglas-Peucker algorithm [13] to each sequence of points in order to obtain the polyline representation (Fig. 4). The Douglas-Peucker algorithm adds vertices to a polyline in order of the distance error until the distance error becomes smaller than the defined threshold. Furthermore, we add vertices until the length of each line segment becomes short than the defined length threshold. This extra process is adopted in order to avoid long thin triangle creation in polygonal model creation step.

Then, we use a vertex of a polyline as a vertex of a polygonal model. Moreover, a line segment is used as a processing element for planar region extraction.

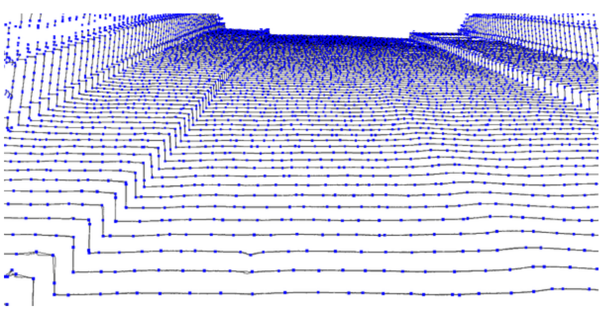

Fig. 4 Polyline approximation of a point cloud.

2.3 Vertex Selection for Polygonal Model from Measured Points

As mentioned in the section 2.1, the point cloud which is obtained by the MMS is often too dense in the direction traversing a road. In the case of using all of measured points, a polygonal model consists of many inadequate thin polygons. Therefore, we should select appropriate points for representing physical objects effectively. The Douglas-Peucker algorithm adds a vertex to a polyline in order of distance error. It means that a vertex of a resultant polyline is placed where the change of the shape is large. Hence, we can select vertices suitable for a polygonal model from a huge amount of measured points by using a vertex of a polyline obtained by the Douglas-Peucker algorithm.

Furthermore, a polygonal model is generally evaluated by the distance from measured points to a polygonal model. The Douglas-Peucker algorithm adds a vertex to a polyline until the maximum distance error is less than a threshold. Hence, the distance error of a polygonal model which uses such a vertex is guaranteed the threshold of the Douglas-Peucker algorithm.

2.4 Estimation of Normal Vector at a Line Segment

In general, the planar region extraction by the region growing approach needs definition of neighborhood elements. And, neighborhood elements are also used to estimate a normal vector. In the case of the traditional point-based method, the data structure like a k-d tree is used for definition of neighborhood points. However, another technique is necessary for the line-based case. Zhang et al. proposed the method which used the cylindrical region whose axis is a query line segment [14]. He et al. also adopted this method to define neighborhood line segments [15]. Zhang's method seems to be reasonable for neighborhoods definition. However, it is time-consuming way, because we have to check all line segments in the previous and the next scanning lines.

There is another way for defining neighborhoods in the line-based case. In the literature [16], the three-dimensional segment tree [17] is used for defining neighborhood line segments. The three-dimensional segment tree determines the relation of inclusion and the relation of partial overlap between intervals in three-dimensional space. The axis aligned bounding box of a line segment can be used for the interval of the three-dimensional segment tree. However, to create the three-dimensional segment tree is very time-consuming task. Moreover, a large amount of memory is also necessary.

In this paper, for fast search of neighborhood line segments, we use the information of laser irradiation angle which is associated with measured points. If two points on consecutive two scanning lines have similar laser irradiation angle, these points are considered to be located near each other. From this observation, we define neighborhood line segments as follows. Let a line segment in the scanning line t be $p_i^t p_j^t$ $(i < j)$, where p_i^t and p_j^t are the end points of a line segment. And, angle θ_i^t and θ_j^t $(0 \leq \theta_i^t, \theta_j^t < 360)$ are assigned to p_i^t and p_j^t, respectively. Here, $p_i^{t+1} p_j^{t+1}$ is defined as the neighborhood of $p_i^t p_j^t$ if the interval of angle $\left[\theta_i^{t+1}, \theta_j^{t+1}\right]$ is included in or intersects with the interval $\left[\theta_i^t - \varepsilon_\theta, \theta_j^t + \varepsilon_\theta\right]$. ε_θ is the margin of angle interval. We perform this process for $t-1$ th, t th and $t+1$ th scanning lines in order to find neighborhoods of a line segment which is included in t th scanning line. Fig. 5 illustrates the definition of neighborhood line segments. In this figure, blue line segments are the neighborhoods of the red line segment.

Fig. 6 shows an example of neighborhoods line segments in three-dimensional space.

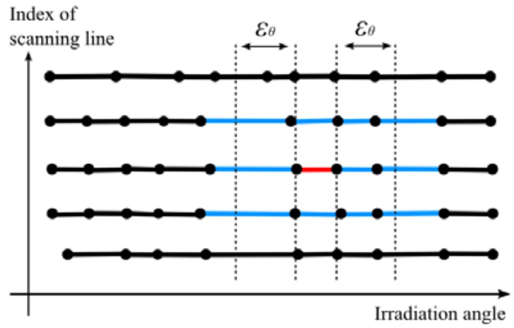

Fig. 5 The definition of neighborhood line segments.

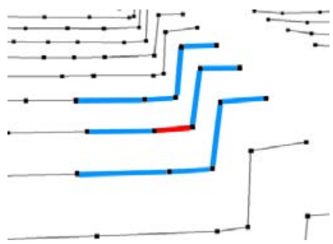

Fig. 6 An example of neighborhood line segments.

In the region growing approach, the difference of the angles of the normal vectors is used for examining whether a neighborhood line segment is added to the region. The least squares fitting to neighborhood elements is often used for estimating a normal vector. However, such a method is not able to derive the precise estimation of a normal vector as mentioned in the section 2.1. Hence, we adopt the local best-fit plane of neighborhood line segments for estimating the normal vector at a line segment. The local best-fit plane is defined as the plane which passes the query line segment and includes the neighborhood line segments most. We use the modified version of the local best-fit plane used in [15] in this paper. We assume that the line segment where the normal vector is calculated is l_i. And, the neighborhood line segments of l_i is represented by $\mathrm{N}(l_i)$. In this situation, a planar surface $P_{i,j}$ is defined by two end points of l_i and the mid-point of $l_j \in N(l_i)$. Then, a neighborhood line segment $l_k = (p_{k1}, p_{k2})$ is defined as an inlier of $P_{i,j}$ if l_k satisfies following two conditions:

1. $\mathrm{d}(p_{k1}, P_{i,j}) \leq \varepsilon_d$ and $\mathrm{d}(p_{k2}, P_{i,j}) \leq \varepsilon_d$,
2. $\overrightarrow{n_{i,j}} \cdot d_k \leq \varepsilon_a$.

Where, $\mathrm{d}(\mathrm{p}, P_{i,j})$ is the perpendicular distance from p to $P_{i,j}$. And, $\overrightarrow{n_{i,j}}$ and d_k are the normal vector of $P_{i,j}$ and the normalized direction vector of l_k, respectively. The first condition represents that l_k is near to $P_{i,j}$ enough. The second represents that l_k is nearly parallel to $P_{i,j}$. Therefore, ε_d and ε_a should be set to small value enough. We calculate the degree of fit as the summation of the lengths of the inlier line segments. Then, the plane with the largest degree of fit is defined as the local best-fit plane P_i of l_i. And, the normal vector $\overrightarrow{n_i}$ of P_i is used as the normal vector at l_i. Fig. 7 shows an example of the local best-fit plane. The green thick line segments are inliers of the plane defined by l_i and l_j. The blue thick line segments are not inlier while these are the neighborhood line segments of l_i. Fig. 8 shows an example of normal vectors estimated by the local best-fit plane. We can find that the normal vector at the line segment around the boundary between a road and a curbstone is precisely estimated.

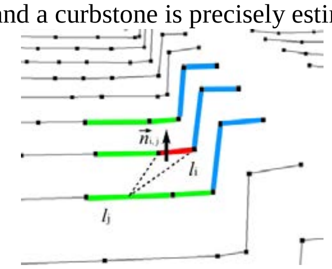

Fig. 7 An example of the local best-fit plane and inliers.

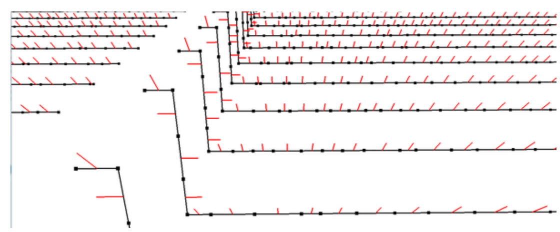

Fig. 8 An example of normal vectors estimated by the local best-fit plane.

3. Acquisition of Boundary Edge Information from a Point Cloud

We apply the region growing method to a set of line segments in order to detect nearly planar regions. In general, the region growing approach starts from selecting a seed as the first query. The query is added to the region and the neighborhoods of the query are examined whether should be added to the region. The conditions of the inlier that was defined in the section 2.4 can be used for deciding whether the neighborhood line segment is added. The neighborhood which is added to the region becomes the next query in order to grow the region. The seed segment is selected by using the degree of fit mentioned in the section 2.4. A large degree of fit means that the target line segment is placed on the large planar region. Thus, we select the seed segment of the largest degree of fit from input line segments that are not assigned to any planar region yet.

After the line-based region growing process, the line segment is labeled with the extracted planar region which the line segment belongs to. A vertex of a line segment is shared by two adjacent line segments. Thus, a vertex has two labels of planar regions which adjacent line segments belong. According to this observation, a vertex at the boundary between different planar regions has two different labels (Fig. 9). We assume that such a vertex is the boundary vertex. We use this information for constraining the boundary edge in the polygonal model creation step.

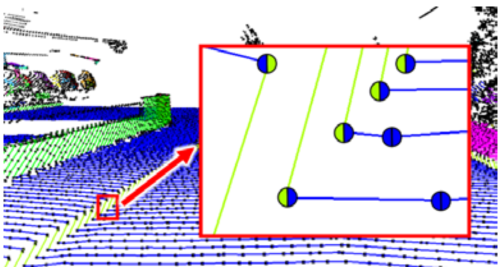

Fig. 9 The label of a line segment and a vertex.

4. Polygonal model creation with boundary edge constraints

4.1 Triangle Strip/Fan Creation

We create a triangular polygonal model by generating triangle strip/fan between consecutive two scan lines. Triangular polygons are created very fast because we find a vertex of a polygon from points included in only two scan lines. This is an advantage of creating triangle strip/fan. We assume that triangle strip/fan is created between t th and $t+1$ th scan lines. The points in each scan line are ordered by the laser irradiation angle. At first, for t th scan line, we set the point which is under the MMS as the start point p_s^t. That is, the start point of t th scan line is the point whose laser irradiation angle is the nearest to $180°$. And, for $t+1$ th scan line, the start point p_s^{t+1} is the point which is the nearest to p_s^t . Then, let the initial query pair of vertex (p_i^t, p_j^{t+1}) be (p_s^t, p_s^{t+1}). Next, for the query pair of vertices (p_i^t, p_j^{t+1}) , the length $|p_i^t p_{j+1}^{t+1}|$ and $|p_{i+1}^t p_j^{t+1}|$ are calculated, respectively. The vertex pair of shorter length is adopted to the vertex of the triangle. That is, the triangle $\{p_i^t, p_{j+1}^{t+1}, p_j^{t+1}\}$ is generated if $|p_i^t p_{j+1}^{t+1}| < |p_{i+1}^t p_j^{t+1}|$. Then, the query pair of vertex is updated to (p_i^t, p_{j+1}^{t+1}). In the opposite case, the triangle $\{p_i^t, p_{i+1}^t, p_j^{t+1}\}$ is generated and the query pair is updated to (p_{i+1}^t, p_j^{t+1}) (Fig. 10). The process mentioned above is the case of ascending order from the start point. We perform the same process in descending order from the start point.

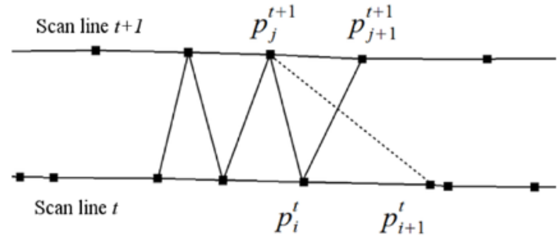

Fig. 10 An example of creating triangle between scan lines.

4.2 Edge Constraints Based on Boundary Edge Information

Triangle creation depends on the length of an edge spanning two scan lines. Hence, an inappropriate edge may be generated at the boundary of the physical object. We introduce edge constraints in the triangle strip/fan creation step in order to prevent an inappropriate edge from being generated. The edge constraint is defined based on the segmentation of planar regions mentioned in the section 3.1. A vertex has two labels of planar regions which adjacent line segments belong. And, a vertex where is located at the boundary of different planar regions has two different labels. During the triangle strip/fan creation step, if both of a vertex pair is a boundary vertex and an identical label appears on the same side of each vertex, such an edge is constrained as the boundary edge. And, the triangle which includes such an edge is generated. If the vertex has been used for a constrained edge once, such a vertex is not used as another constrained edge. Fig. 11 shows an example of the constrained edge.

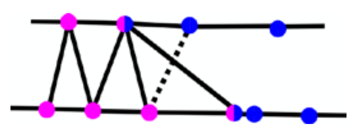

Fig. 11 An example of the constrained edge.

5. Experimental results

5.1 Experiments by the Simulated Point Cloud

In actual, a point cloud obtained by a MMS does not have the topological information of a polygonal model. Therefore, we are not able to evaluate the accuracy of boundary edge information by comparing with the actual mobile mapping data. Hence, we created the polygonal model of the scene around a road including the curbstone for evaluating the proposed method. The surface of a road and curbstones have the red and the green color materials, respectively. And, the outside of a road has the blue color material. Then, we use the software, BlenSor [18], for the MMS simulation. The BlenSor is able to simulate several types of laser scanning. We selected the types of the laser scanning similar to the MMS and moved the source point of the laser irradiation along the trajectory on the road. Fig. 12 shows the result of the MMS simulation. The laser irradiation starts from vertical upward direction. Thus, the measured point which is associated with $180°$ means just under the source of laser irradiation. After the MMS simulation, we obtained approximately 1.76 million points. Average interval of measured points along the direction traversing the road is 2mm. On the other hand, average distance between scanning lines is 60cm. Each point has the color information at the location where the point is measured. Fig. 13 shows the obtained point cloud with color information. We use such a point cloud for evaluating the precision of the boundary information obtained by the proposed method.

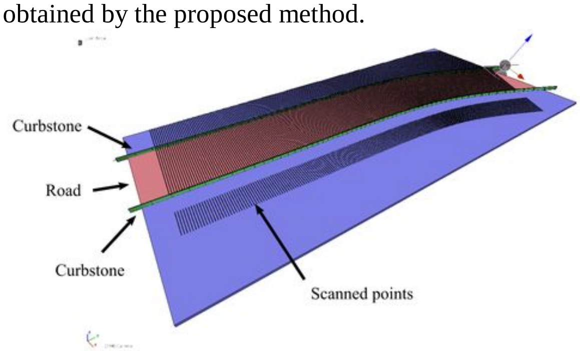

Fig. 12 The simulation of MMS by the BlenSor.

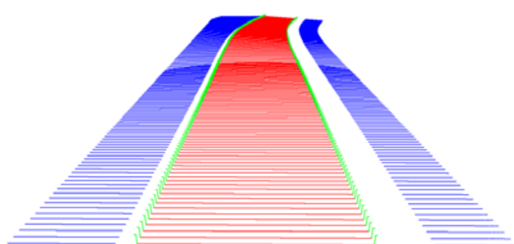

Fig. 13 The point cloud by the MMS simulation.

At first, we evaluate the planar region extraction for obtaining the boundary edge information. Figs. 14, 15 and 16 show the visual comparison between the traditional point-based region growing and the line-based region growing. Fig. 14 shows the result of road region detection by using the traditional point-based region growing. We use the points which have nearly $180°$ of laser irradiation angle as the seed points. Fig. 15 shows the result which uses another parameters. On the other hand, Fig. 16 shows

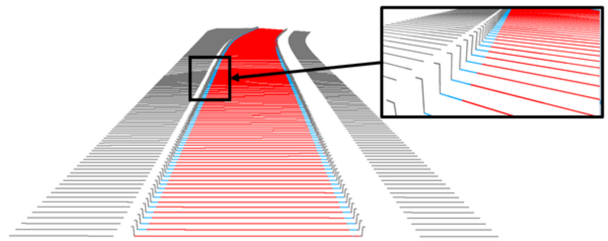

Fig. 14 The result of road extraction by the point-based region growing.

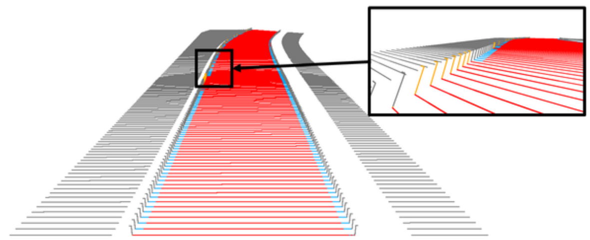

Fig. 15 The result of the point-based region growing using another parameters.

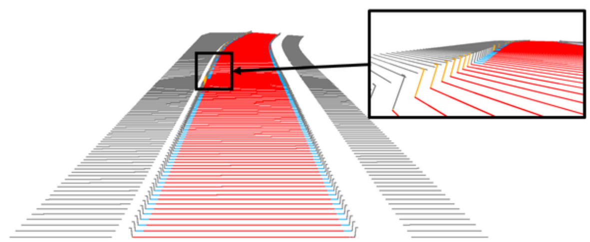

Fig. 16 The result of road extraction by the proposed method.

the result of road detection by the line-based method. Table 1 shows parameters for the road detection.

We assume that the point is correct if the point of the actual road region is extracted. Such a point is colored in red in Figs. 14, 15 and 16. The false positive point is the extracted point that is not the road region. And, the false negative is the point of the road region which is not extracted. The false positive point and the false negative point are colored in orange and cyan, respectively. Fig. 14 shows that the region growing process is stopped in front of the curbstone. Fig. 15 shows the case of which the angle threshold for the region growing is slightly larger than Fig. 14. In the case of Fig. 15, several parts of the extracted region are expanded to the curbstone. This result indicates that it is difficult to distinguish a road region from other regions by a simple point-based region growing approach because the change of the normal vector is gradual in the vicinity of the boundary between the road region and the curbstone.

Table 1 Parameters for the region growing.

	Fig. 14	Fig. 15	Fig. 16
Distance error for the Douglas-Peucker algorithm	-	-	0.05m
Maximum length of a line segment	-	-	0.3m
Radius of neighborhood	0.75m	0.75m	-
Margin for neighborhood (ε_g)	-	-	5°
Difference of height (ε_d)	0.05m	0.05m	0.05m
Difference of normal vector (ε_θ)	20°	30°	30°

Table 2 shows the statistical comparison of the point-based method and the proposed method. This result indicates that the proposed method is able to precisely extract the planar region which represents the road region. Gradual change of normal vector direction is caused by the normal vector estimation which uses neighborhood points including different planar regions. In contrast, the proposed method estimates the normal vector by only of neighborhood line segments which are on the plane including the target line segment. Consequently, the direction of the normal vector at the line segment on the road region is greatly different from other planar regions. Thus, the road region can be distinguished from other planar regions precisely by the line-based region growing.

Table 2 The statistical comparison between the point-based method and the proposed method.

	Fig. 14	Fig. 15	Fig. 16
Correct	1,355,458	1,368,056	1,414,377
False Positive	9	10,622	0
False Negative	58,919	46,321	0

Next, we evaluate the effectiveness of the polygonal model creation with boundary edge constraints. We create the polygonal model by the method mentioned in the section 4 (Fig. 17). And, the polygonal model without the boundary edge constraints is also created for comparison (Fig. 18). In visual evaluation, we can find that several inadequate edges are generated at the boundary between the road and the curbstone in the polygonal model created without boundary edge constraints as shown in Fig. 18.

We calculate the Hausdorff distance between the original road model and the created polygonal model. We use the Meshlab software [19] in order to calculate the Hausdorff distance. Figs. 19 and 20 show a part of the sampling points on the polygonal model. The sampling points are colored according to the Hausdorff distance to the original road model shown in Fig. 12. Dark color shows large Hausdorff distance at the sampling point. Vertices of the created polygonal model are a part of the measured input points. Therefore, all of vertices of the created polygonal model are on the surface of the original model. This means that the Hausdorff distance at the vertex of the polygonal model is zero in this case. On the other hand, the Hausdorff distance becomes large at a sampling point on an edge of a polygonal model if the edge is inappropriate for representing the shape of the object. Table 3 shows the statistics of the calculated Hausdorff distance. The total number of sampling points is 150,000. In the case of the polygonal model created without edge constraints, the maximum is 0.101m and approximately 900 points have the value more than 0.005m. This means that some inadequate edges appeared in the polygonal model. On the other hand, the maximum value is 0.003m in the case of the polygonal model created by the proposed method. This means that the topological information of the polygonal model is generated adequately and the boundary edges appeared precisely.

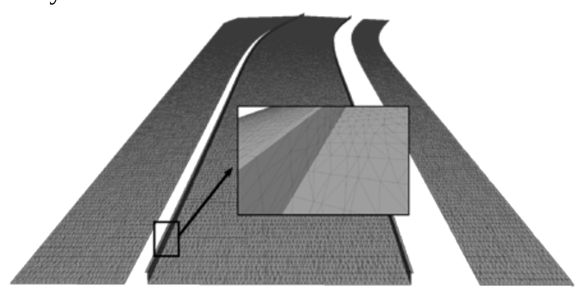

Fig. 17 The polygonal model created by the proposed method.

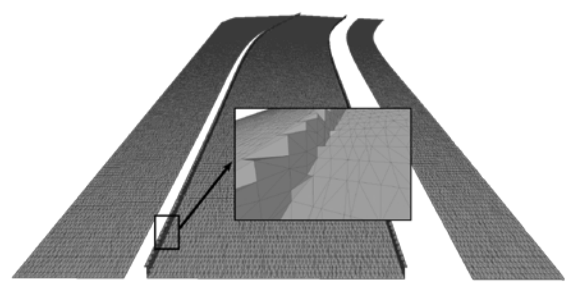

Fig. 18 The polygonal model without boundary edge constraints.

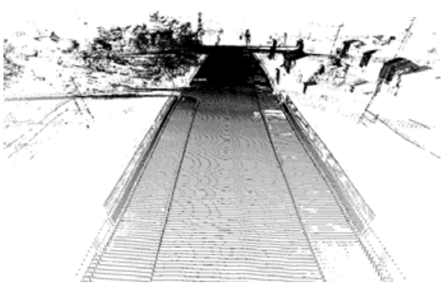

Fig. 21 An actual point cloud data.

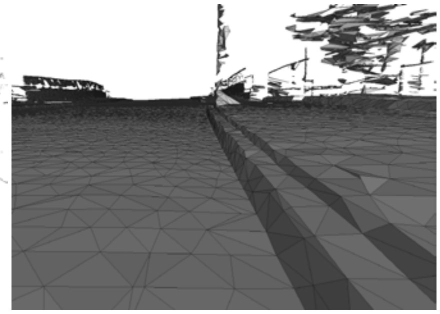

Fig. 22 The polygonal model created by the proposed method from an actual point cloud data.

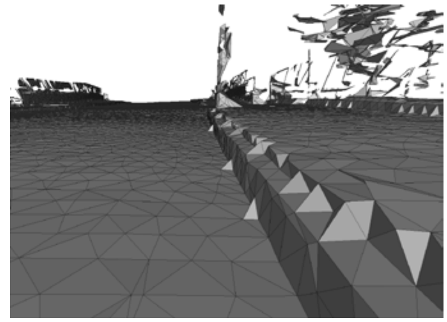

Fig. 23 The polygonal model which is created without boundary edge constraints.

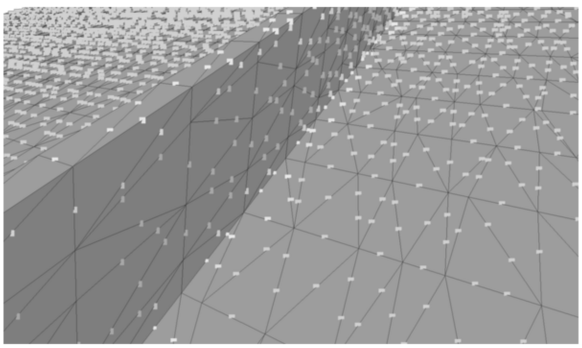

Fig. 19 Sampling points on the polygonal model created by the proposed method.

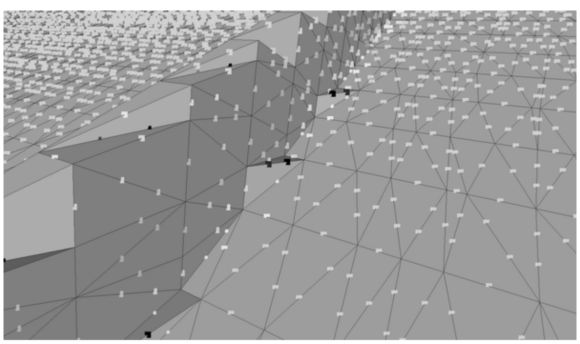

Fig. 20 Sampling points on the polygonal model without boundary edge constraints.

Table 3 The statistics of Hausdorff distance.

Hausdorff distance (m)	# of sampling points	
	The proposed method	The method without constraints
0.0 ~ 0.005	150,000	149,111
0.005 ~ 0.01	0	17
0.01 ~ 0.05	0	420
0.05 ~	0	452

5.2 Experiments by the Actual Point Cloud

Next, we apply the proposed method to actual point cloud data. And also, the polygonal model is created without boundary edge constraints for comparison. Fig. 21 shows an actual point cloud data. We create the polygonal model by the proposed method from this point cloud data. Fig. 22 shows the magnified view of the polygonal model in the vicinity of the boundary between the road and the sidewalk. Fig. 23 is the view same as the Fig. 22 of the polygonal model which was created without boundary edge constraints for comparison. In the case of the polygonal creation without boundary edge constraints, boundary edges are jagged. However, boundary edges are appropriately represented in the polygonal model by the

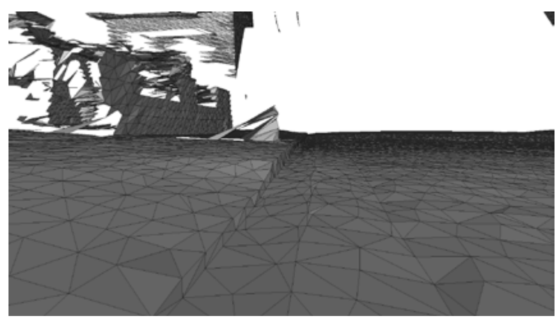

Fig. 24 Another view of the polygonal model by the proposed method.

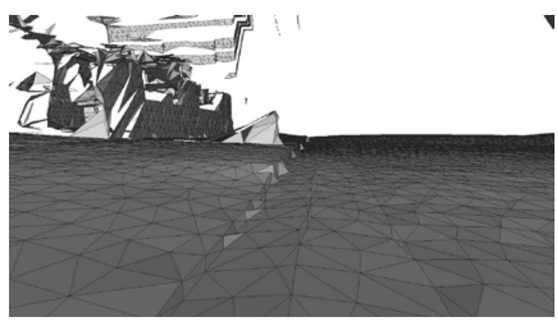

Fig. 25 The same view as Fig. 24 of the polygonal model without boundary edge constraints.

proposed method. Figs. 24 and 25 show another view of the polygonal models. In this view, the shape of low step is included. The height of the low step is approximately 5 cm. The proposed method is able to represent the shape of the low step by the appropriate boundary edges. On the other hand, some portions of low step are degenerated in the case of the polygonal model without boundary edge constraints.

6. Conclusion

In this paper, we proposed the method for creating the polygonal model that is able to express appropriate boundary edge. The boundary edge is defined as the joint part of planar regions in the proposed method. Then, appropriate connection of the boundary vertices is constrained in the triangle creation step.

We are able to obtain the cross-section view very easily and precisely from the polygonal model representing the appropriate boundary edge. This is useful for the maintenance of a road and a curbstone. Moreover, to obtain the precise road boundary can be used as the information for autonomous driving technology. As the future work, we consider estimation method for road parameters that include width of road, cross slope, curvature of a curve, based on precisely extracted road polygonal model.

References

[1] K. Ishikawa, Y. Amano, T. Hashizume, J. Takiguchi, and N. Kajiwara, “A Mobile Mapping System for Precise Road Line Localization Using a Single Camera and 3D Road Model,” Journal of Robotics and Mechatronics, vol. 19, no. 2, pp. 174–180, 2007.

[2] K. Aoki, K. Yamamoto, and H. Shimamura, “Evaluation Model for Pavement Surface Distress on 3d Point Clouds from Mobile Mapping System,” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 87–90, July, 2012.

[3] J. Demantke, B. Vallet, and N. Paparoditis, “Streamed Vertical Rectangle Detection in Terrestrial Laser Scans for Facade Data Production,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. I-3, pp. 99–104, July, 2012.

[4] J. Demantke, B. Vallet, and N. Paparoditis, “Facade Reconstruction with Generalized 2.5d Grids,” ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-5,pp. 67–72, 2013.

[5] K. Kato, H. Seki, and M. Hikizu, “3-D Obstacle Detection Using Laser Range Finder with Polygonal Mirror for Powered Wheelchair,” International Journal of Automation Technology, Vol. 9, No. 4, pp 373–380, 2015.

[6] A. Jaakkola, J. Hyyppa, H. Hyyppa, and A. Kukko, “Retrieval Algorithms for Road Surface Modelling Using Laser-Based Mobile Mapping,” Sensors, Vol. 8, No. 9, pp. 5238–5249, 2008.

[7] C. Mertz, “Continuous Road Damage Detection Using Regular Service Vehicles,” Proceedings of the ITS World Congress, Oct., 2011.

[8] F. Pauling, M. Bosse, and R.Zlot, “Automatic Segmentation of 3D Laser Point Clouds by Ellipsoidal Region Growing,” Proceedings of Australasian Conference on Robotics and Automation, pp. 11–20, Dec., 2009.

[9] T. Watanabe, T. Niwa, and H. Masuda, “Registration of Point Clouds from Terrestrial and Portable Laser Scanners,” International Journal of Automation Technology, Vol. 10, No. 2, pp. 163–171, 2016.

[10] H. Yasuda, K. Taketomi, K. Tachi and M. Suzuki, ” Study on Method to Draw Cross Sectional Profiles of Rivers Using Laser Scanned Ground Height Data (in Japanese),” Advances in River Engineering, 9, pp. 247–252.

[11] D. Belton and K.-H. Bae, “Automating Post-Processing of Terrestrial Laser Scanning Point Clouds for Road Feature Surveys,” Proceedings of the ISPRS Commission V Mid-Term Symposium on Close Range Image Measurement Techniques, Vol. XXXVIII, pp. 74–79, June, 2010.

[12] J. He and H. Masuda, “ An Efficient Surface Generation Method for Vehicle-Based Mobile Mapping System,” Proceedings of the International Conference on Civil and Building Engineering Informatics, pp.31-38, Nov., 2013.

[13] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, 1973.

[14] Z. Zhang and O. D. Faugeras, “Finding clusters and planes from 3D line segments with application to 3D motion determination,” Lecture Notes in Computer Science Vol. 588, pp. 227–236, 1992.

[15] Y. He, C. Zhang, and C. Fraser, “A Line-Based Spectral Clustering Method for Efficient Planar Structure Extraction from Lidar Data,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-5/W2, pp. 103–108, Oct, 2013.

[16] R. Miyazaki, M. Yamamoto, E. Hanamoto, H. Izumi, and K. Harada, “A Line-Based Approach for Precise Extraction of Road and Curb Region from Mobile Mapping Data,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-5, pp. 243–250, May, 2014.

[17] M. d. Berg, M. v. Kreveld, M. Overmars, and O. Schwarzkopf, ” Computational Geometry: Algorithms and Applications,” pp. 231–237, Springer-Verlag, 1997.

[18] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, “BlenSor: Blender Sensor Simulation Toolbox”, In Advances in Visual Computing, pp. 199–208, Springer Berlin Heidelberg, 2011.

[19] MeshLab, “an open source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes,” http://meshlab.sourceforge.net/, accessed Mar. 7. 2016.

Ryuji Miyazaki received the B.S. and M.S. degrees in 1996 and 1998, respectively. And, He received Ph.D in 2010 from Hiroshima University. He is a lecturer of the Faculty of Psychological Science. His current research is mainly in the area of the digital geometry processing. He is especially interested in the application of a large scale point cloud processing. He is a member of IEEE and Japan Society for Precision Engineering.

Makoto Yamamoto is an engineer of Sanei Co.. His main work is the development of CG and CAD system for designing highways and general roads. Currently, he engages in the research and development of the 3D travel orbit used in comfortable riding and autonomous vehicle driving in the view of economical vehicle control.

Koichi Harada is a visiting professor of the Graduate School of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and Ph.D in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research topics include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is also a professor emeritus of Hiroshima University.