

Adaptive Segmentation of Large-Scale Anisotropic Point-Clouds Captured by Mobile Mapping Systems

Yuma Mori¹ , Keisuke Kohira² and Hiroshi Masuda³

¹ The University of Electro-Communications, <u>yuma.mori@uec.ac.jp</u> ² The University of Electro-Communications, <u>keisuke.kohira@uec.ac.jp</u> ³ The University of Electro-Communications, <u>h.masuda@uec.ac.jp</u>

Corresponding author: Hiroshi Masuda, h.masuda@uec.ac.jp

ABSTRACT

A mobile mapping system (MMS) is effective for capturing dense point-clouds of roads and roadside objects. In order to create 3D models from huge point-clouds, it is necessary to efficiently extract objects from point-clouds. However, since points captured using the MMS are highly anisotropic, it is difficult to detect local connectivity between points using a constant threshold. In this paper, we discuss the method to define adaptive thresholds for local connectivity of highly anisotropic point-clouds captured using the MMS. In our method, point-clouds are mapped on the 2D lattice and they are connected on the lattice. Then we introduce adaptive thresholds by simulating laser scanning with the MMS and comparing the simulated point intervals with actual ones. By using the adaptive thresholds, continuous surfaces can be stably extracted from large-scale point-clouds.

Keywords: Point-Cloud, Mobile Mapping System, Surface Extraction **DOI:** https://doi.org/10.14733/cadaps.2019.466-477

1 INTRODUCTION

A mobile mapping system (MMS) is an equipment on which as laser scanners, cameras, GPSs, and an IMU are mounted. Fig. 1. shows the MMS that we used to capture point-clouds. The MMS determines the own positions and attitudes using the GPS and IMU, and measures 3D coordinates on objects by emitting laser beams. The MMS typically outputs point-clouds and the trajectory of the laser scanner. Each point has a 3D coordinate, an intensity value and a GPS time. The GPS time is the time sent from the satellite and represents when the point was captured. The trajectory of the laser scanner is represented as a series of 3D coordinates with GPS time.

Point-clouds captured using the MMS are useful to create 3D models, such as buildings [19], roads [9], pole-like objects [15], and so on. In recent years, the performance of the laser scanner has been greatly improved. The laser scanners on the MMS can measure from 300,000 to 1 million points per second. When the MMS measures points for several hours, billions of points can be obtained. In order to create 3D models from huge point-clouds, it is necessary to efficiently and

robustly segment point-clouds and extract surfaces. If surfaces can be detected, object shapes are reconstructed combining surfaces.

However, points are highly anisotropic because the intervals of points depends on the vehicle speed, the setting of the laser scanner, and the distances from the scanner positions. Fig. 2. shows an example of points captured using the MMS. In this case, the point intervals in the traveling direction of the vehicle are ten times or larger than the scanning direction of the laser beam. For processing point-clouds captured using MMS, it is required to cope with highly anisotropic point-clouds.

When primitive surfaces, such as planes and cylinders, are extracted from point-clouds, it is necessary to estimate the local connectivity of points. So far, many researchers have proposed methods for estimating local connectivity of point-clouds. However, most of them are not suitable to process highly anisotropic point-clouds. Local geometric features based on eigenvalues of the principal component analysis (PCA) are often used for segmenting point-clouds into each surface [5; 24]. For calculating local geometric features, it is very important to adequately determine neighborhood sizes. When point-clouds are isotropic, a fixed neighborhood size can be used and neighbor points can be detected using a sphere with the fixed radius [11], or *k*-closest points with fixed k [12]. On the other hand, when point-clouds are anisotropic, the optimal neighborhood has to be adaptively estimated for each point. Pauly, et al. estimated neighborhood sizes using the local sampling density and the local curvature [17]. Demantke proposed the dimensionally-based scale selection method, in which local geometric features are calculated using multiple neighborhood sizes is selected using Shannon entropy [3]. These methods are useful when



Figure 1: Mobile mapping system: (a) Vehicle with an MMS, (b) Close-up of MMS, (c) Laser scanner.



Figure 2: Anisotropic point-clouds captured by an MMS.

adequate neighborhood sizes depending on the distances from the scanner position. However, pointclouds captured using the MMS, point-density is largely different in the scanning direction and the travelling direction at each point, as shown in Fig. 2. In such cases, existing methods do not work well, because they assume that points are locally isotropic.

The region growing is also often used for segmenting point-clouds. Voxel-based region growing [4; 20; 25] or octree-based region growing [23] can be used for point-clouds. Random sample consensus (RANSAC) [6] is also popular for segmenting point-clouds [22]. Since this method is robust to noises, it is often used for noisy point-clouds [21; 26]. These methods also require thresholds for local connectivity to extract surfaces, but it is difficult to adaptively estimate thresholds of local connectivity for highly anisotropic points.

For fixed terrestrial laser scanners (TLS), local connectivity can be estimated using the azimuth and latitude angles of laser beams. Masuda, et al. used this approach for region growing [13] and the RANSAC method [14]. Since laser beams are spherically emitted at the equal angle intervals from the source of the laser beam, each point-cloud can be projected onto 2D lattice in angle space. In these methods, the threshold for local connectivity was adaptively estimated depending on the distances from the laser scanner. Although this method is very effective and stable, it cannot be applied for MMS data, because the scanner position on the MMS is always moving.

The method for the fixed TLS is based on the simulated laser scanning. To extend this method to MMS, it is necessary to simulate laser scanning for MMS. Simulated laser scanning is mainly used for artificial point-clouds. Gschwandtner, et al. implemented a tool for simulating point-clouds for fixed TLSs with various parameters [8]. Fukano, et al. simulated point-clouds to be captured by a MMS and created training data for classification [7]. However, these methods were not developed for estimating local connectivity.

In this paper, we propose a new method for estimating local connectivity of point-clouds captured using a MMS. Our method is based on 2D mapping and simulated point-clouds for the MMS. First, neighborhood of points are efficiently and robustly detected using point-clouds and the parameters of the laser scanner. Then we simulate the laser beam emitted from the MMS and estimate the local connectivity of points. Once local connectivity of points are estimated using our method, points can be segmented using existing methods, such as region growing and RANSAC methods.

2 OVERVIEW

2.1 Point-Clouds Measured with the MMS

In this paper, we process point-clouds captured by the MMS. We suppose that the MMS outputs point-clouds and the trajectory of the laser scanner. Since the vehicle position is measured using GPSs, IMUs, and odometers, the trajectory of the laser scanner can be obtained from the vehicle position using the relative position of the laser scanner on the vehicle. The trajectory data contain 3D coordinates, positions and attitudes, and GPS times of the laser scanner. The coordinates are represented in the geodetic reference system, such as WGS84.

Point-clouds contain a sequence of points with 3D coordinates, intensity values, and GPS times. Since the original coordinates from a laser scanner are represented in the scanner-centered coordinate system, they are transformed to coordinates based on the geodetic reference system by using the positions and attitudes of the laser scanner.

Point-clouds in our examples were measured using the Mitsubishi MMS-X [18], as shown in Fig. 1. The laser scanner mounted on this MMS is RIEGL VQ 250. The laser scanner measures 3D coordinates with laser beams with 360 degrees, as shown in Fig. 3(a). Then, there are two important parameters that affect point density. One is the rotation frequency, which indicates how many times the laser beam rotates per second. The other is the scan rate, which indicates how many measurements are made per second. The rotational frequency and the scan rate of RIEGL VQ 250 is 100Hz and 300,000 measurements, respectively. Since the direction of the laser beam rotates at

a constant rotation frequency, the laser beam is spirally irradiated when the vehicle moves forward, as shown in Fig. 3(b).

The reason why point-clouds become highly anisotropic can be explained by the two parameters



Figure 3: Trajectory of laser beams: (a) Direction of laser beams, (b) Trajectory of laser beam directions.

and the vehicle speed. In the case of RIEGL VQ 250, the interval between consecutively scanned points can be calculated as about 1 cm at 5 m ahead because the laser beam rotates $2\pi/3,000$ radian in the time interval of point sampling. On the other hand, the intervals in the travelling direction are 11 cm when the vehicle travels at 40 km/h, which is 11 m/sec.

2.2 Process for Estimating Local Connectivity

Fig. 4. shows a process of our method. First, point-clouds are mapped on the 2D lattice using GPS times and scanner parameters. The laser beam rotates at a constant speed, and coordinates are sampled at regular time intervals. Therefore, we can convert each coordinate captured using the MMS onto (I, J) on the 2D image. Then, neighbor points are obtained on the 2D image.

For estimating local connectivity, distance thresholds are necessary to determine whether neighbor points are connected or not. Two neighbor points are connected only when the distance is smaller than a threshold. However, adequate thresholds depend on the scanner position, the velocity of the vehicle, and the scanner parameters, and the geometric property of target objects. Therefore, we simulate point clouds by supposing neighbor points were captured from continuous surfaces, and compare them with actual point-clouds. If the distance between actual neighbor points is much larger than the simulated distance, the assumption is discarded and they are not connected. In our method, the threshold is adaptively calculated for each pair of neighbor points, and local connectivity is estimated. Then point-clouds are segmented into continuous surfaces according to obtained local connectivity.

3 MAPPING POINTS ONTO 2D LATTICE

We project point-clouds onto the 2D lattice using GPS time and the parameters of laser scanners. Masuda, et al. projected point-clouds captured using the MMS onto a 2D plane for two types of laser scanners [16], and detected neighborhood using the Delaunay triangulation. Bruno, et al. projected point-clouds onto a 2D lattice using the discrete rotation angles and the rotation frequency [2]. Kohira, et al. formulated the mapping onto the 2D lattice using the rotation frequency and the pulse repetition frequency [10], and proposed image-based compression for point-clouds. Bruno and Kohira derived neighbor points on the 2D lattice, but they did not discuss local connectivity.

In this paper, we use 2D mapping proposed in [10] using the laser scanner that spirally emits laser beams, such as Fig. 3(a). Let f be the rotation frequency, N be the scan rate, and ω be the pulse repetition frequency. The pulse repetition frequency ω is defined as N/f.

Since the laser beam rotates once every 1/f second, points are grouped into segments every 1/f second, as shown in Fig. 5. We call points in each segment as a scan-line. As shown in Fig. 5, the adjacent scan-line includes points measured 1/f second later. In order to map point-clouds onto the 2D lattice, the phase number and rotation number are assigned to each point, as shown in Fig. 6. The phase number indicates the order of measurement on each scan-line, and the rotation number is the sequential number of a scan-line.



Figure 4: Process for surface detection.

Then the phase number *I* and the rotation number *J* can be calculated using the following equations.

$$I = \operatorname{int} (\omega \cdot \operatorname{fmod} (t, 1/f))$$

$$J = \operatorname{int} (f \cdot t)$$
(3.1)
(3.2)

fmod(x, y) computes the floating-point remainder of x/y. t is the elapsed time from the start of the measurement and it is obtained as the difference in GPS time between the current point and the initial point.

Point-clouds are converted to the 2D lattice by mapping each point to (I,J). Fig. 7 shows an example of projected points. Blue pixels indicate that no points were mapped at the pixels.



Figure 5: The nearest point adjacent scan-line.



Figure 6: The phase number and the rotation number.



Figure 7: Mapping onto the 2D lattice: (a) Point-clouds, (b) points on the 2D lattice.

4 SIMULATED POINT-CLOUDS

4.1 Scanner Position and Laser Direction

We represent a point-cloud as $\{\mathbf{p}_{i,j}\}\$ and $\{t_{i,j}\}\$, where $\mathbf{p}_{i,j}$ is a 3D coordinate mapped at (i,j) on a point image, and $t_{i,j}$ is the GPS time. We also represent the trajectory of the scanner positions as 3D coordinates $\{\mathbf{q}_k\}\$ and GPS times $\{u_k\}$.

For assigning a scanner position to point $\mathbf{p}_{i,i}$, the following index k is detected.

$$u_k \le t_{i,j} < u_{k+1} \tag{4.1}$$

Then the scanner position $\mathbf{s}_{i,j}$ is calculated as:

$$\mathbf{s}_{i,j} \equiv \mathbf{s}(t_{i,j}) = \frac{(u_{k+1} - t_{i,j})\mathbf{q}_k + (t_{i,j} - u_k)\mathbf{q}_{k+1}}{u_{k+1} - u_k}$$
(4.2)

The rotation axis of the laser beam can be calculated using vectors $\mathbf{p}_{i,j} - \mathbf{s}_{i,j}$. We project points on scan-line *j* on a unit sphere, and fit a plane to the projected points. The normal vector of the plane is estimated as the rotation axis of the laser scanner. We denote the rotation axis of scan-line *j* as \mathbf{a}_j .

In Fig. 8, the laser beam every 0.2 seconds is shown in red, and the trajectory of the laser scanner is shown in yellow.



Figure 8: Laser beams emitted from MMS.

4.2 Estimation of Connectivity

The normal direction of each point can be roughly estimated by fitting a plane to neighbor points. Neighbor points are first selected as points in a circle with radius r_{2D} on 2D lattice. Although point intervals in 3D space are largely different in the *I* and *J* directions, the nearly same number of points are selected in the both directions. Then we eliminate points outside the sphere with radius r_{3D} in 3D space. The normal vector is estimated for each point. We denote the estimated normal vector of $\mathbf{p}_{i,j}$ as $\mathbf{n}_{i,j}$. In this paper, we specified $r_{2D} = 4$ pixels and $r_{3D} = 30$ cm.

Then we consider whether neighbor points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+k,j+l}$ are on the same continuous surface. We suppose that k and l are small integers. We denote the rotation matrix that rotates θ around the axis \mathbf{a}_j as $\mathbf{R}_j(\theta)$. Since the rotation angle between the two points are $2\pi k/\omega$, the laser beam direction for $\mathbf{p}_{i+k,j+l}$ can be estimated as:

$$\mathbf{R}_{j}\left(\frac{2\pi k}{\omega}\right)(\mathbf{p}_{i,j}-\mathbf{s}_{i,j}) \tag{4.3}$$

We denote that the normalized direction as $\bar{\mathbf{v}}_{i+k,j+l}$. The laser beam for $\mathbf{p}_{i+k,j+l}$ can be simulated as a straight line with the direction $\bar{\mathbf{v}}_{i+k,j+l}$ through the scanner position $s_{i+k,j+l}$.

We suppose that $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+k,j+l}$ are on the same plane whose normal vector is $(\mathbf{n}_{i,j} + \mathbf{n}_{i+k,j+l})/2$. Then the position of $\mathbf{p}_{i+k,j+l}$ can be simulated as the intersection point between the plane and the simulated laser beam, as shown in Fig. 9. We denote the estimated position as $\overline{\mathbf{p}}_{i+k,j+l}$.

If the assumption that the two points are on the same plane is correct, the actual distance $|\mathbf{p}_{i,j} - \mathbf{p}_{i+k,j+l}|$ will be nearly equal to the simulated distance $|\mathbf{p}_{i,j} - \overline{\mathbf{p}}_{i+k,j+l}|$. Therefore, we estimate that $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+k,j+l}$ are on the same surface only if:

$$|\mathbf{p}_{i,j} - \mathbf{p}_{i+k,j+l}| < \lambda |\mathbf{p}_{i,j} - \overline{\mathbf{p}}_{i+k,j+l}|$$
(4.4)

 λ is a constant greater than 1. This value is experimentally determined depending the MMS. In our evaluation, we specified $\lambda = 1.3$.



Figure 9: The simulated position of a neighbor point.

5 SURFACE EXTRACTION

5.1 Out-of-Core Processing

When the MMS measures 300,000 to 1 million points per second, the data size becomes too large to be loaded on RAM when the MMS travels for several hours. Therefore, it is necessary to segment point-clouds for processing on common PCs. In our method, point-clouds are mapped on the 2D lattice, in which the width of the lattice is fixed, but the height increases according to the travelling time of the vehicle. Since the connectivity of neighbor points can be locally determined, point-clouds can be divided and processed in an out-of-core manner.

Fig. 10 shows out-of-core processing for huge point-clouds. Since points are stored in files sequentially, they are loaded from a hard disk in order of GPS time. Therefore, we load a certain number of files and maintain the points for processing, as shown in Fig. 10(b). When processing of points in a file is completed, the points are removed from RAM and the next file is loaded.

For segmenting buildings and other roadside objects, it is desirable to remove roads so that each object is represented as a connected component of points. Several methods have been proposed for detecting roads [9]. In our case, it is easy to detect roads, because the trajectory of the vehicle is given and it is known that the roads exists directly under the vehicle, as shown in Fig. 10(c). Points are regarded as roads when the Z coordinates are nearly equal to or less than the road height. Fig. 10(d) shows road points extracted from points in Fig. 10(b).



(a) A part of point-clouds

(b) Loading of point-clouds





(c) Detection of road height (d) Detection of points on roads Figure 10: Out-of-core processing for huge point-clouds.

5.2 Segmentation into Continuous Surfaces

Continuous surfaces are detected by connecting neighbor points that satisfy Eqn. (4.4). In out-ofcore processing, continuous surfaces are updated each time a new file is loaded. If any point is not added to a continuous surface, the detection of the surface is completed. Fig. 11 shows detected continuous surfaces in different colors. The result shows that out method works adequately.

Continuous surfaces can be further segmented to planes if necessary. In Fig. 12, a surface is subdivided into planar regions using the RANSAC method.







Figure 12: Segmentation into planar regions.

5.3 Experimental Results

We evaluated our method using point-clouds measured in a Japanese urban area. In the MMS we used, points are stored in files every 300 thousand points. In our experiments, points were processed every 3 million points in an out-of-core manner.

In our method, point-clouds are mapped on the 2D lattice. Therefore, local connectivity can be obtained by determining a threshold for the distance between neighbor points on the 2D lattice. First, we adaptively specified using our method and connected neighbor points if Eqn. (4.4) is satisfied. Fig. 13. shows connected components.

For comparison, we extracted connected components using fixed thresholds. In Fig. 14, the thresholds were specified as 8cm, 16 cm, and 24 cm. In Fig. 14(a), many fragmented regions were generated. In Fig. 14 (b), front walls were adequately extracted, but some regions were fragmented. In Fig. 14(c), points were excessively connected. The results show that adequate local connectivity requires adaptive thresholds because thresholds depend on the distance from the laser scanner, the irradiation angles, and the vehicle speed.

CPU time was measured using a PC with 3.3GHz Intel Core i9 and 64GB RAM. In this evaluation, point-clouds with 3 million points were converted into a mesh model based on local connectivity. The calculation time is shown in Tab. 1. CPU time on mesh generation includes generation of triangular mesh models by connecting neighbor points on the lattice. This result shows that our method can process point-clouds in a practical time.



Figure 13: Points connected based on our method: (a) Point-clouds, (b) Connected points.



Figure 14: Points connected using a fixed threshold *d*: (a) d = 8 cm, (b) d = 16 cm, (c) d = 24 cm.

Process	CPU Time
Loading Point-Cloud	4.1 sec
Mapping onto Lattice	0.04 sec
Position and Velocity	0.07 sec
Connecting Points	8.3 sec
Generation of Continuous Surfaces	3.8 sec

 Table 1: CPU Time for processing 3 million points.

6 CONCLUSION

In this paper, we proposed a method for deriving local connectivity of points and segmenting largescale anisotropic points into connected surfaces. We mapped points onto a 2D lattice using GPS times and scanner parameters, and introduced adaptive thresholds by simulating laser scanning with the MMS. By using adaptive thresholds, the connectivity of points could be obtained for anisotropic point-clouds.

In future work, we would like to use this method for automatically detecting and classifying various objects from very large point-clouds. In addition, we would like to improve performance of our method.

Yuma Mori, <u>http://orcid.org/0000-0003-2172-6612</u> *Keisuke Kohira*, <u>http://orcid.org/0000-0002-7692-0386</u> *Hiroshi Masuda*, <u>http://orcid.org/0000-0001-9521-6418</u>

REFERENCES

- [1] Boulaassal, H.; Landes, T.; Grussenmeyer, P.; Tarsha-Kurdi, F.: Automatic Segmentation of Building Facades Using Terrestrial Laser Data, ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, 65-70, 2007.
- [2] Bruno, V.; Mathieu, B.; Andrés, S.; Beatriz, M.; Nicolas, P.: TerraMobilita/iQmulus Urban Point Cloud Analysis Benchmark, Computers & Graphics, 49, 126-133, 2015. <u>http://dx.doi.org/10.1016/j.cag.2015.03.004</u>
- [3] Demantke, J.; Mallet, C.; David, N.; Vallet, B.: Dimensionality Based Scale Selection in 3D Lidar Point Clouds, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-5/W12, 97-102, 2011. <u>http://dx.doi.org/10.5194/isprsarchives-XXXVIII-5-W12-97-2011</u>
- [4] Deschaud, J. E.; Goulette, F.: A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing, The 5th International Symposium on 3D Data Processing, Visualization and Transmission, 2010.
- [5] Dittrich, A; Weinmann, M.; Hinz, S.: Analytical and Numerical Investigations on the Accuracy and Robustness of Geometric Features Extracted from 3D Point Cloud Data, ISPRS Journal of Photogrammetry and Remote Sensing, 126, 195–208, 2017. <u>http://dx.doi.org/10.1016/j.isprsjprs.2017.02.012</u>
- [6] Fischler, M. A.; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, 381-395, 1981. <u>http://dx.doi.org/10.1016/B978-0-08-051581-6.50070-2</u>
- [7] Fukano, K., Masuda, H.: Classification for Roadside Objects Based on Simulated Laser Scanning, The International Conference on Civil and Building Engineering Informatics, 2015.
- [8] Gschwandtner, M., Kwitt, R., Uhl, A. and Pree, W.: BlenSor: Blender Sensor Simulation Toolbox, International Symposium on Visual Computing, pp. 199-208, 2011. <u>http://dx.doi.org/10.1007/978-3-642-24031-7_20</u>

- [9] He J.; Masuda H.: Reconstruction of Roadways and Walkways Using Point-Clouds from Mobile Mapping System, Asian Conference on Design and Digital Engineering, 100099. 2012.
- [10] Kohira, K.; Masuda H.: Point-Cloud Compression for Vehicle-Based Mobile Mapping System Using Portable Network Graphic, SPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Commission VI/ WG VI/4, 2017. <u>http://dx.doi.org/10.5194/isprs-annals-IV-2-W4-99-2017</u>
- [11] Lee, I.; Schenk, T.: Perceptual Organization of 3D Surface Points. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV-3A, 193–198, 2002.
- [12] Linsen, L.; Prautzsch, H.: Local versus Global Triangulations, EUROGRAPHICS, 2001. http://dx.doi.org/10.2312/egs.20011021
- [13] Masuda, H.; Tanaka, I.: Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds, Computer-Aided Design and Applications, 6(3), 2009, 387-398. <u>http://dx.doi.org/10.3722/cadaps. 2009.387-398</u>
- [14] Masuda, H.; Niwa, T.; Tanaka, I.; Matsuoka, R.: Reconstruction of Polygonal Faces from Large-Scale Point-Clouds of Engineering Plants, Computer-Aided Design and Applications, 12(5), 2015, 555-563. <u>http://dx.doi.org/10.1080/16864360.2015.1014733</u>
- [15] Masuda, H., Oguri, S., He, J.: Shape Reconstruction of Poles and Plates from Vehicle based Laser Scanning Data, Informational Symposium on Mobile Mapping Technology, 2015.
- [16] Masuda, H.; He, J.: TIN Generation and Point-Cloud Compression for Vehicle-Based Mobile Mapping Systems, Advanced Engineering Informatics, 29(4), 841-850, 2015. <u>http://dx.doi.org/10.1016/j.aei.2015.05.007</u>
- [17] Mitra, N. J.; Nguyen, A.; Guibas, L.: Estimating Surface Normals in Noisy Point Cloud Data. International Journal of Computational Geometry Applications, 14, 261-276, 2004. <u>http://dx.doi.org/10.1142/S0218195904001470</u>
- [18] Mobile Mapping System High-accuracy GPS Mobile Measurement Equipment, <u>http://www.mitsubishielectric.com/bu/mms/</u>Mitsubishi Electric.
- [19] Nan, L.; Sharf, A.; Zhang, H.; Cohen-Or, D.; Chen, B.: SmartBoxes for Interactive Urban Reconstruction, Transactions on Graphics, 29(4), Article 93, 2010. <u>http://dx.doi.org/10.1145/1833349.1778830</u>
- [20] Nurunnabi, A.; Belton, D.; West, G.: Robust Segmentation in Laser Scanning 3D Point Cloud Data, International Conference on Digital Image Computing Techniques and Applications, 1-8, 2012. <u>http://dx.doi.org/10.1109/DICTA.2012.6411672</u>
- [21] Pauly, M.; Keiser, R.; Gross, M.: Multi-scale Feature Extraction on Point-Sampled Surfaces, Computer Graphics Forum, 22(3), 281-289, 2003. <u>http://dx.doi.org/10.1111/1467-8659.00675</u>
- [22] Schnabel, R.; Wahl, R.; Klein, R.: Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum, 26(2), 214-226, 2007. <u>http://dx.doi.org/10.1111/j.1467-8659.2007.01016.x</u>
- [23] Vo, A. V.; Truong-Hong, L.; Laefer, D. F.; Bertolotto, M. .: Octree-Based Region Growing for Point Cloud Segmentation, ISPRS Journal of Photogrammetry and Remote Sensing, 104, 88-100, 2015. <u>http://dx.doi.org/10.1016/j.isprsjprs.2015.01.011</u>
- [24] Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C.: Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105, 286-304, 2015. <u>http://dx.doi.org/10.1016/j.isprsjprs.2015.01.016</u>
- [25] Xiao, J.; Zhang, J.; Adler, B.; Zhang, H.; Zhang, J.: Three-Dimensional Point Cloud Plane Segmentation in Both Structured and Unstructured Environments, Robotics and Autonomous Systems, 61(12), 1641-1652, 2013. <u>http://dx.doi.org/10.1016/j.robot.2013.07.001</u>
- [26] Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. .: Investigation on the Weighted Ransac Approaches for Building Roof Plane Segmentation from Lidar Point Clouds. Remote Sensing, 8(1), 5, 2015. <u>http://dx.doi.org/10.3390/rs8010005</u>