

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Point Cloud Clustering Using Panoramic Layered Range Image

---

Masafumi Nakagawa

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.76407>

---

## Abstract

Point-cloud clustering is an essential technique for modeling massive point clouds acquired with a laser scanner. There are three clustering approaches in point-cloud clustering, namely model-based clustering, edge-based clustering, and region-based clustering. In geoinformatics, edge-based and region-based clustering are often applied for the modeling of buildings and roads. These approaches use low-resolution point-cloud data that consist of tens of points or several hundred points per  $m^2$ , such as aerial laser scanning data and vehicle-borne mobile mapping system data. These approaches also focus on geometrical knowledge and restrictions. We focused on region-based point-cloud clustering to improve 3D visualization and modeling using massive point clouds. We proposed a point-cloud clustering methodology and point-cloud filtering on a multilayered panoramic range image. A point-based rendering approach was applied for the range image generation using a massive point cloud. Moreover, we conducted three experiments to verify our methodology.

**Keywords:** point-cloud clustering, point-based rendering, terrestrial laser scanning, surface extraction, 3D edge extraction

---

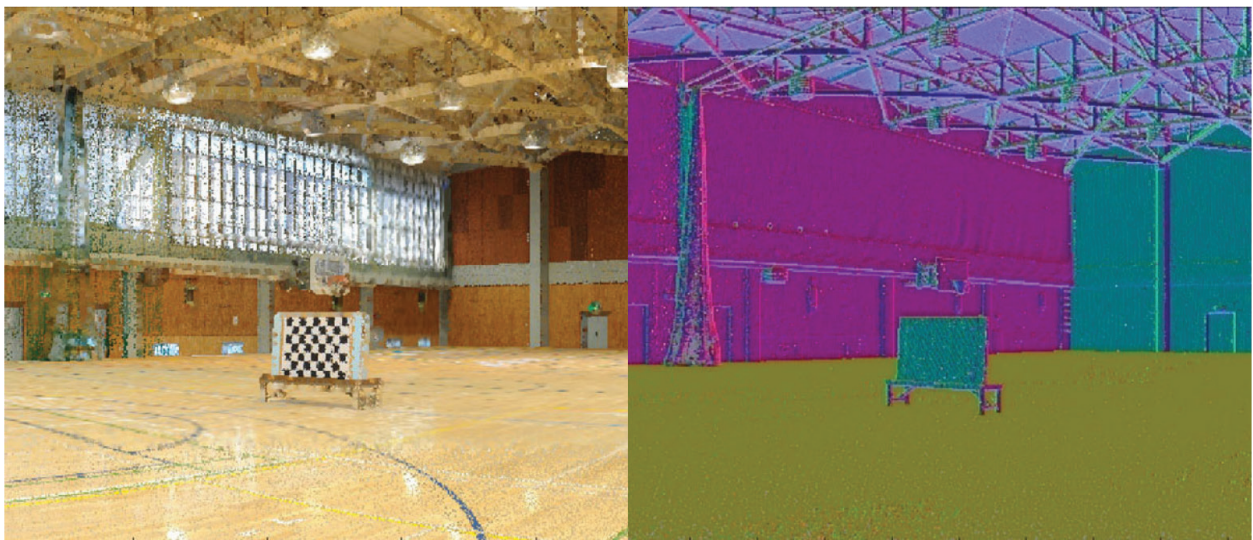
## 1. Introduction

Massive point-cloud acquisition is an effective approach for 3D modeling of unknown objects in various fields, such as urban mapping, indoor mapping, plant management, factory management, heritage documentation, and infrastructure asset inspection and management. In construction fields, base maps and 3D data are required for managing processes of construction, maintenance, rehabilitation, and replacement. Online maps, such as Google Maps and

OpenStreetMap, are useful for approximate construction surveys in urban areas. However, online maps are often insufficient for infrastructure inspection to recognize the details of natural features. Thus, base maps and 3D data should be prepared before inspection. Massive point-cloud data can be acquired with a terrestrial laser scanner, mobile mapping systems (MMSs), handheld laser scanners, and cameras using structure from motion (SfM) methodology. SfM is a methodology for reconstructing a scene using multiple cameras simultaneously from all available relative motions through key point detection, feature matching, motion estimation, triangulation, and bundle adjustment. In an open sky environment, aerial photogrammetry and SfM using an unmanned aerial vehicle (UAV) are more effective than ground-based scanning. On the other hand, when environments include natural obstacles, such as trees, a terrestrial laser scanner is more effective than a UAV or MMSs. In indoor navigation and building information modeling (BIM), floor maps and 3D data are also required. We expected terrestrial laser scanners and indoor MMSs to be adequate for colored point-cloud acquisition in an indoor environment.

Moreover, point-cloud clustering is an essential technique for modeling massive point clouds. **Figure 1** shows an example of point-cloud clustering using a terrestrial laser scanner data acquired in an indoor environment.

There are three clustering approaches in point-cloud clustering, namely model-based clustering [1], edge-based clustering [2], and region-based clustering [3]. Model-based clustering is a 3D model preparation approach. The model-based clustering requires 3D models such as CAD models to estimate simple objects or point clusters from the point cloud. In 3D industrial modeling, standardized objects, such as pipes, boxes, and parts, are prepared as CAD models in advance. Although the model-based clustering is suitable for modeling known objects such as the standardized objects, the model-based clustering is unsuitable for modeling unknown objects such as complex and natural objects. On the other hand, in modeling unknown objects, such as buildings and roads in geoinformatics and civil engineering



**Figure 1.** Point-cloud clustering: colored point cloud (left image) and clustered point cloud (right image).

fields, edge-based and region-based clustering are often applied [4]. These approaches use low-resolution point-cloud data that consist of tens of points or several hundred points per  $\text{m}^2$ , such as aerial laser and vehicle-borne MMSs data. These approaches also focus on geometrical knowledge [5] and 2D geometrical restrictions, such as the depth from a platform [6] and discontinuous point extraction on each scanning plane from the MMSs [7] to extract features. In urban areas and indoor environments, although there are simple features consisting of lines and planes, there are many complex features consisting of curved lines and surfaces with unclear boundaries. Moreover, point-cloud data are generally acquired with a terrestrial and mobile laser scanner from many viewpoints and view angles for 3D modeling. Like the conventional approaches, range image processing is proposed to apply 2D restrictions with an interactive procedure in 3D plant modeling. However, viewpoints for range image rendering are limited to data acquisition points.

Thus, our aim was to improve region-based point-cloud clustering in modeling after point-cloud integration. We also focused on region-based point clustering to extract a polygon from a massive point cloud, because it is not easy to estimate accurate edges from point clouds acquired with a laser scanner. In region-based clustering, random sample consensus (RANSAC) [8] is a suitable approach for surface detection and estimation. However, local work space should be selected to improve performance in a surface estimation from a massive point cloud. Moreover, it is hard to determine whether a point lies inside or outside a surface with conventional RANSAC.

In this chapter, we first proposed a point-cloud clustering methodology on a panoramic layered range image generated with point-based rendering from a massive point cloud. Next, we conducted three experiments to verify our methodology. The first experiment was a 3D edge and surface extraction for indoor modeling using an indoor MMS. The second experiment was a 3D edge and surface extraction for 3D bridge modeling using a terrestrial laser scanner. The third experiment was a 3D edge and surface extraction for ground surface and feature extraction using a terrestrial laser scanner. Even though the acquired data had low homogeneity of spatial point density, these experiments confirmed that a terrestrial laser scanner could cover complex surfaces, including flat surfaces, slopes, and steps. We also confirmed that our proposed methodology could achieve point-cloud clustering to extract these features from complex environments.

## 2. Methodology

Our proposed processing flow for point-cloud clustering is shown in **Figure 2**. First, we register and integrate point-cloud data acquired from a viewpoint. Next, the point-cloud data are projected into the image space with translation, view angle, and resolution parameters in “Panoramic multilayered range image generation with point cloud rendering” to generate several range images. Then, normal vectors around each projected point are estimated using the 3D coordinate values of the point cloud in “Normal vector estimation in panoramic multilayered range image.” Next, edges are extracted from depth images generated in the panoramic

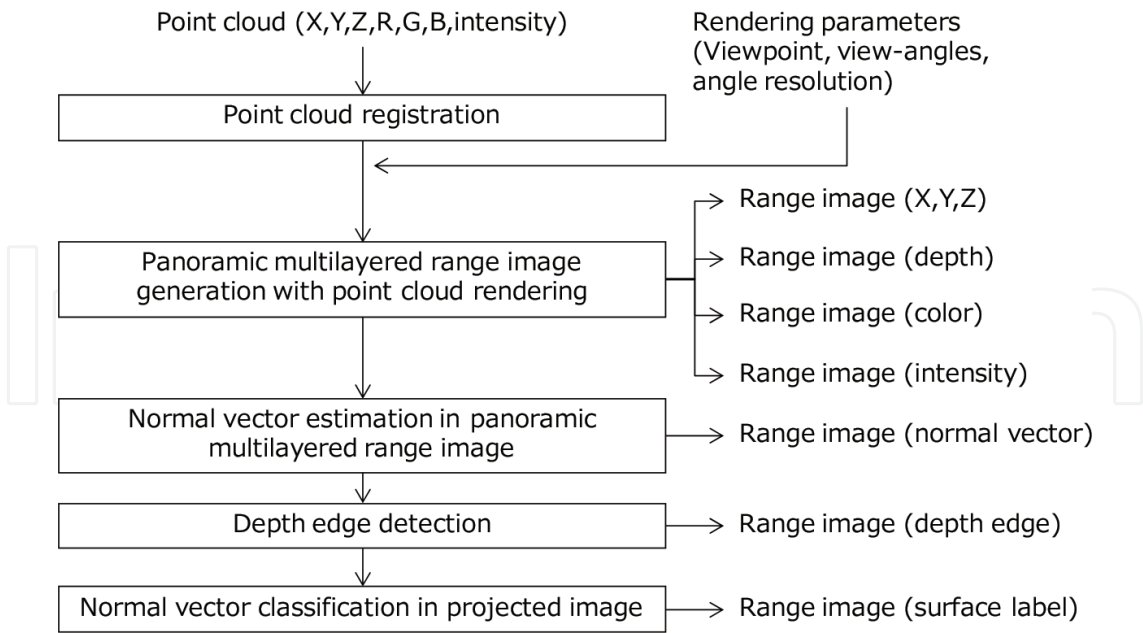


Figure 2. Processing flow for point-cloud clustering.

multilayered range image generation in “Depth edge detection.” Finally, groups that have similar direction in the point cloud are extracted after normal vector classification in the projected image in “Normal vector classification in projected image.” Generated range images are managed in a multilayered range image.

2.1. Panoramic multilayered range image generation with point-cloud rendering

An advantage of 3D point-cloud data is that they allow accurate display from an arbitrary viewpoint and 3D modeling. Additionally, point-cloud data have the potential for applications such as panoramic image geo-referencing and distance value-added panoramic image processing for 3D geographical information system (GIS) visualization [9, 10]. However, point-cloud visualization has two technical issues. The first issue is the near-far effect caused by distance differences from the viewpoint to scanned points. The second issue is the transparency effect caused by rendered hidden points. These effects degrade the quality of point-cloud visualization. Thus, we focus on methodologies to improve the quality of point-cloud visualization. The Splat-based ray tracing [11] can generate a photorealistic curved surface from point-cloud data; surface generation requires the long period in the 3D work space. Moreover, the curved-surface description is unsuitable for representing urban objects such as CAD and GIS data. Therefore, we have applied a point-based rendering and filtering, which we call a layered image-based depth arrangement refiner for versatile rendering (LIDAR VR) [12] for point-cloud rendering.

The processing flow of LIDAR VR is described as follows. First, the sensors acquire a point cloud with additional color data such as RGB data. The sensor position is defined as the origin point in a 3D work space. Second, a multilayered range image from the simulated viewpoint is generated using the point cloud. Finally, the generated multilayered range image is filtered



to generate missing points in the rendered result using distance values between the view-point and objects. The colored point cloud is projected from a 3D space to a panorama space. This transformation simplifies viewpoint translation, filtering, and point-cloud browsing. The LIDAR VR data consist of a projection model and multilayered range image, as shown in **Figure 3**. The panorama model can be selected from a spherical, cylindrical, plane, hemi-spherical, or cubic model. In this chapter, the spherical model is described. First, the measured point data are projected onto a spherical surface as a range of data. Next, the measured point data are projected onto a spherical surface to manage X, Y, Z, R, G, B, and the intensity values in the multilayered panorama space. Then, azimuth and elevation angles are calculated using 3D vectors generated from the viewpoint and the measured points. The azimuth and elevation angles are converted to panorama image coordinate values with adequate spatial angle resolution in the range data. Finally, a spherical panorama image is generated from the measured point cloud.

Based on this panorama projection, the multilayered range data with a translated viewpoint are generated using the point cloud, as shown in **Figure 4**. When a panorama space is generated using points from  $P_1$  to  $P_{10}$  from a viewpoint  $X_o$  the points from  $P_1$  to  $P_{10}$  are continuously

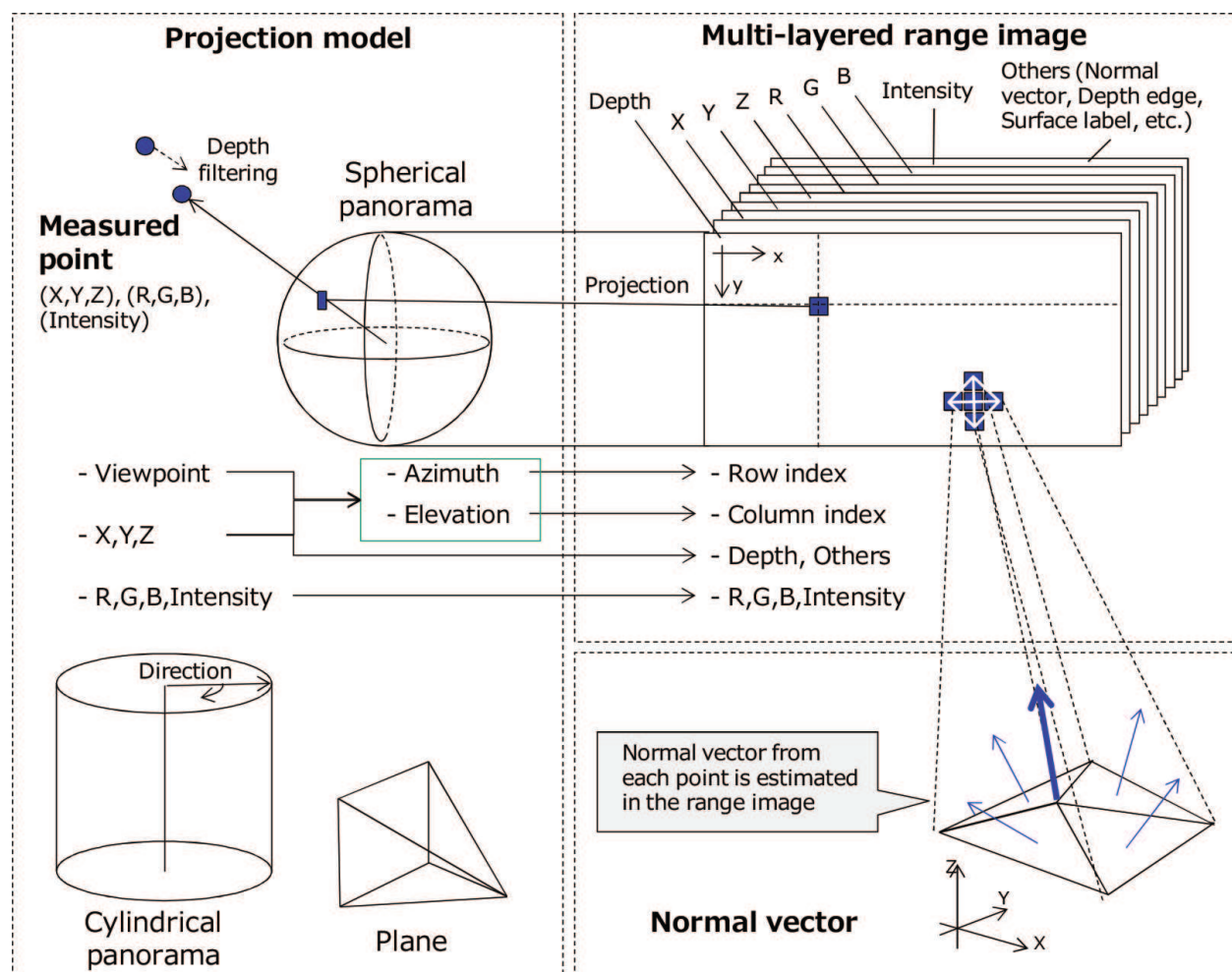
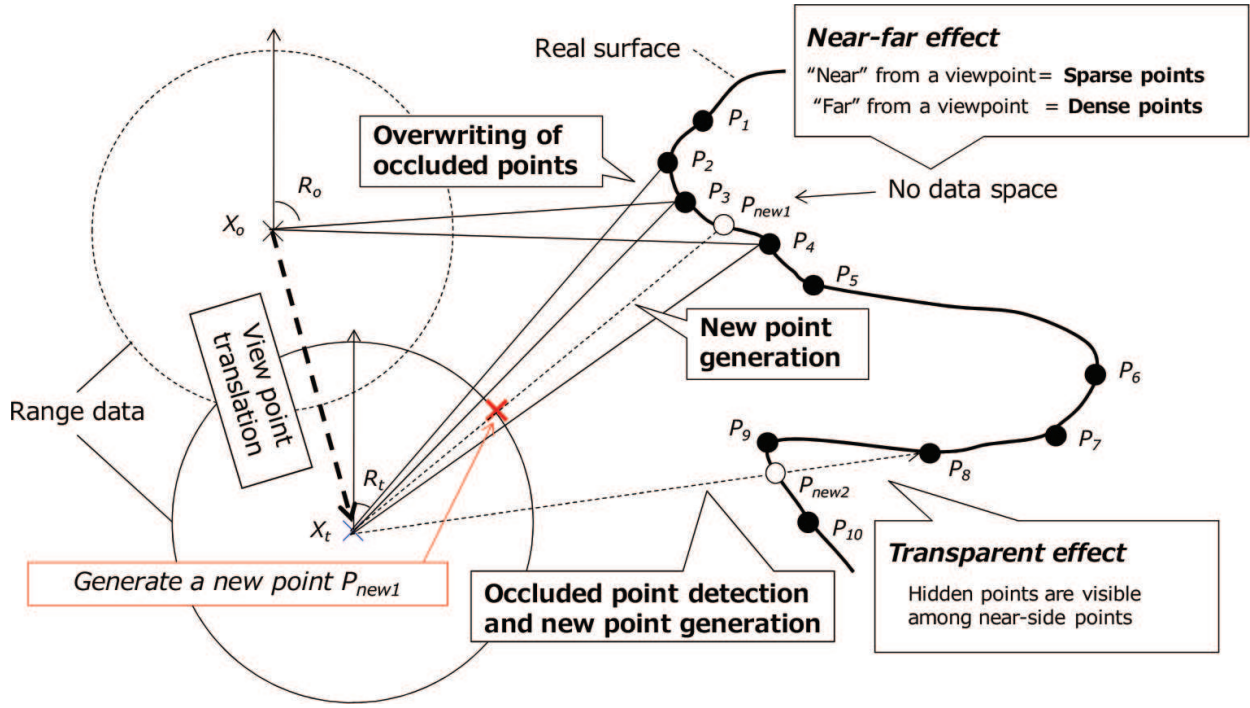


Figure 3. LIDAR VR data model.



**Figure 4.** Point distribution calculated by viewpoint translation in range data, occlusion detection using the point cloud, and point-cloud interpolation with distance information.

arranged in the range data. An azimuth or elevation angle from a viewpoint  $X_o$  to a measured point  $P_1$  is denoted as  $R_o$ . On the other hand, when a panorama space is generated using the points from  $P_1$  to  $P_{10}$  from a different viewpoint  $X_t$ , the angle from the viewpoint  $X_t$  to the measured point  $P_1$  is denoted as  $R_t$ . Thus, the change in angle from  $R_o$  to  $R_t$  affects the arrangement of the projected points in the range data.

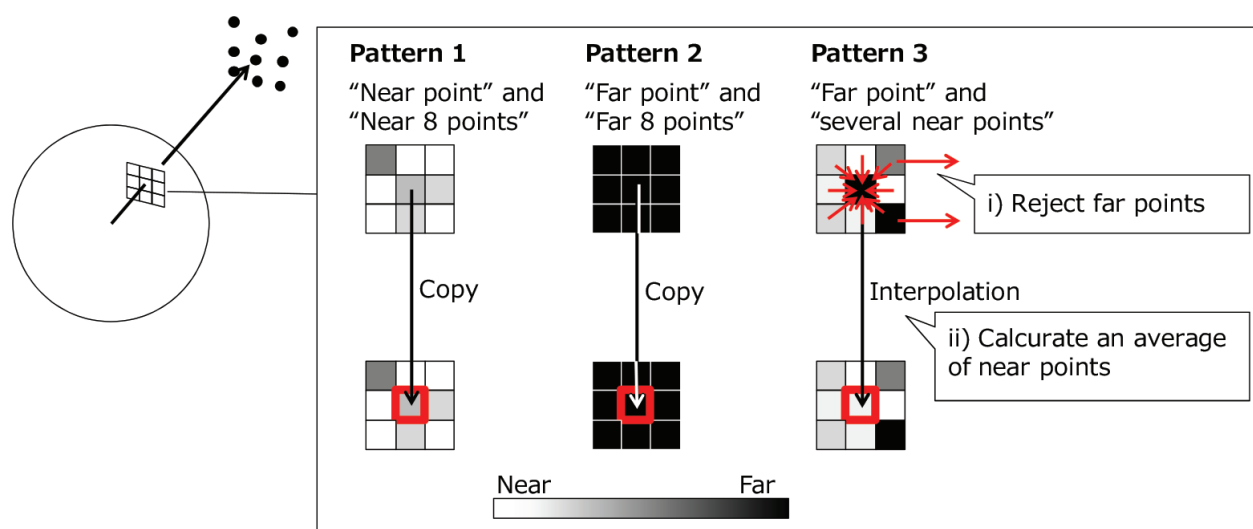
After the viewpoint translation, three types of filtering are applied to point-cloud rendering, as shown in **Figure 4**. The first filtering is the occluded point overwriting. **Figure 4** shows an example to overwrite the projected point  $P_1$  by the projected point  $P_2$ . After the viewpoint translation from  $X_o$  to  $X_t$ , the projected point  $P_1$  becomes an occluded point behind  $P$ . Thus, **Figure 4** shows that  $P_1$  is overwritten by  $P_2$ . The second filtering is the new point generation in the no-data space. **Figure 4** also shows an example to generate a new point  $P_{new1}$ . After the viewpoint translation from  $X_o$  to  $X_t$ , a no-data space occurs between the projected points  $P_3$  and  $P_4$ . Therefore, **Figure 4** shows that  $P_{new1}$  is generated between the projected points  $P_3$  and  $P_4$ . The third filtering is the occluded point replacement. **Figure 4** shows an example to replace an occluded point  $P_8$  with a new point  $P_{new2}$ . After the viewpoint translation from  $X_o$  to  $X_t$ , the point  $P_8$  is visible between points  $P_9$  and  $P_{10}$ . However, the point  $P_8$  exists behind the real surface. Thus, the occluded point  $P_8$  should be given a new distance value as  $P_{new2}$ . The new distance value is calculated the distance values of points  $P_9$  and  $P_{10}$  through the pixel-selectable averaging filter developed in this study, which we now describe.

In general image processing, each pixel value in an image is resampled by using pixel values around it when the image is transformed. A similar technique is applied to the pixel-selectable

averaging filter to improve the quality of the range data generated from point-cloud data with the view-point translation. However, general image resampling techniques, such as the nearest interpolation, linear interpolation, and cubic interpolation, degrade the quality of the range data because the resampling blends various data, such as valid points, occluded points, measurement noises, and missing data. Therefore, the pixel-selectable averaging filter is applied to this technical issue. The pixel-selectable averaging filter extracts valid points around a point for the resampling, as shown in **Figure 5**. This processing consists of a detection of valid data extraction and rejection of occluded points, noises, and missing data, and missing-point regeneration.

The detailed flow of the pixel-selectable averaging filter is described as follows. First, a three-by-three block of pixels is prepared in the range data projected from point clouds. The center point in the block is the focus point in the range data. Second, the block is checked to see whether valid points exist. When there are more than two valid pixels, the processing moves to the next step. If there is only the focus point, it is deleted as spike noise. When the focus point is a missing part, a new pixel value such as color and intensity value is given to the focus point using the other valid points around the center point in the block. Third, after these point extraction steps, an average value of valid points in the block within the search range is calculated to overwrite the focus point value. The average value is a distance from the viewpoint to the valid points. This processing is applied to each channel in the RGB and intensity image. However, when the center point in the block has a distance value within the search range, the overwriting processing is not performed. Because, when the point can be defined approximately as the nearest surface, the overwriting processing has a possibility of degrading geometrical accuracy and image quality in this case. This processing sequence is applied to all points.

In the valid point extraction, a range of search distances should be given. The distance from the viewpoint to the nearest point found among the extracted points is defined as the start



**Figure 5.** Pixel-selectable averaging filter.



value of the search range. Moreover, the start value plus a defined distance parameter is assumed as the end value. The defined distance parameter is determined with the continuity of the points in the point cloud. For example, the defined parameter would be between 10 cm and 1 m from experience, when trees and building walls are measured.

Thus, the pixel-selectable averaging filter uses valid points in the range data to achieve an interpolation without reducing geometrical accuracy by a uniform smoothing effect. **Figure 6** shows an example of processing result.

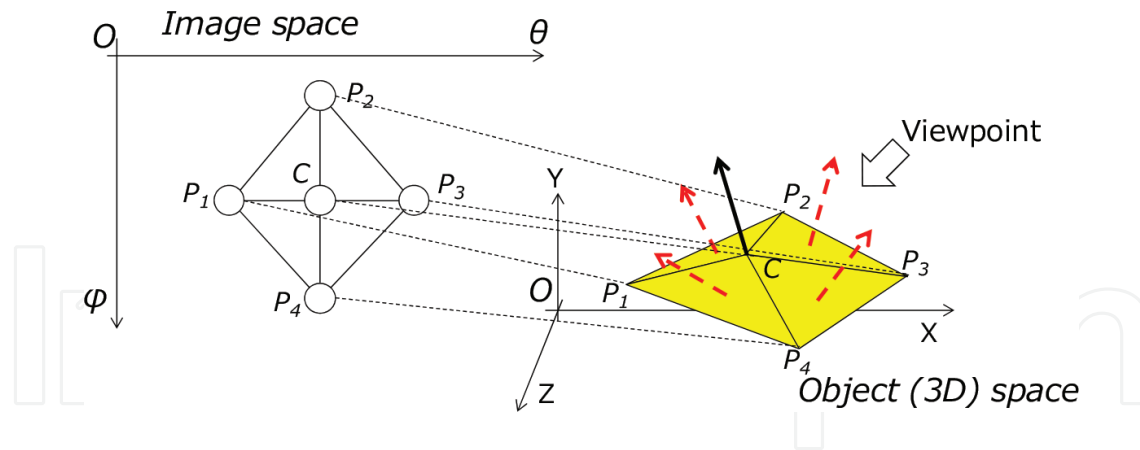
**2.2. Normal vector estimation in a panoramic multilayered range image**

A normal vector can be estimated using three points in point cloud with a triangle patch or mesh generation processing. In 2D image processing, the Delaunay division is a popular algorithm. The Delaunay division can also be applied for 3D point-cloud processing with millions of points [13]. However, using the Delaunay division, it is hard to generate triangle patches for more than hundreds of millions of points without a high-speed computing environment [14, 15]. Thus, we focused on point-cloud rendering that restricts visible point-cloud data as a 2D image. A closed point detection and topology assignment can be processed as 2D image processing.

In our normal vector estimation, four faces in a range image are generated to estimate normal vectors of a point in point cloud, as shown in **Figure 7**. First, a point in the projected point cloud on a panoramic layered range image is defined as point C. Next, the projected points  $P_1, P_2, P_3,$



**Figure 6.** LiDAR VR processing result: input point cloud (upper image) and output point cloud (bottom image).



**Figure 7.** Normal vector estimation in panoramic multilayered range image.

and  $P_4$  in the range image are set from point  $C$  with  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$  pixels in vertical and horizontal directions. Triangulation is applied to these points as vertexes  $C-P_1-P_2$ ,  $C-P_2-P_3$ ,  $C-P_3-P_4$ , and  $C-P_4-P_1$  with a clockwise topology in the image space. Moreover, parameters  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$  in this procedure depend on the accuracy and resolution of the measurement data taken from the laser scanner or stereo camera. When the accuracy and resolution are high enough, these parameters are set as one pixel. These parameters are set to more than one pixel for low accuracy and resolution measurement data to keep a smooth condition of normal vectors on a flat surface. This procedure, which is based on 2D image processing, can provide a higher topology attachment to the point cloud.

Additionally, the normal vector on each triangle is estimated using the 3D coordinate values of each point. When five points consisting of a center and four vertex points exist on the same plane in 3D space, each normal vector has the same direction. When point  $C$  exists on the edge of the 3D space, two clusters can be classified by two directions. Moreover, when point  $C$  exists on the corner of the 3D space, each triangle has a different direction. Surfaces, edges, and corners in the 3D space were estimated in point-cloud data using these clues. In this research, we used the point cloud taken from a laser scanner that presents difficulties for measuring edges and corners clearly. Thus, the average value of each normal vector is used as a normal vector of point  $C$ . These procedures were iterated to estimate the normal vectors of all points in point cloud.

### 2.3. Normal vector-based point clustering

Point clusters are generated from a classification result of normal vectors. The accuracy of point-cloud classification can be improved with several approaches such as the Mincut, Markov network, and fuzzy-based algorithms [16–18]. However, in this study, we focused on verifying the practicality of our point-based rendering for point-cloud clustering. Thus, we applied multilevel slicing as a simple classification algorithm to classify normal vectors, as shown in **Figure 8**.

This classification detected boundaries of point clusters with the same normal vectors. Moreover, clustered normal vectors were compared with normal vectors of neighboring

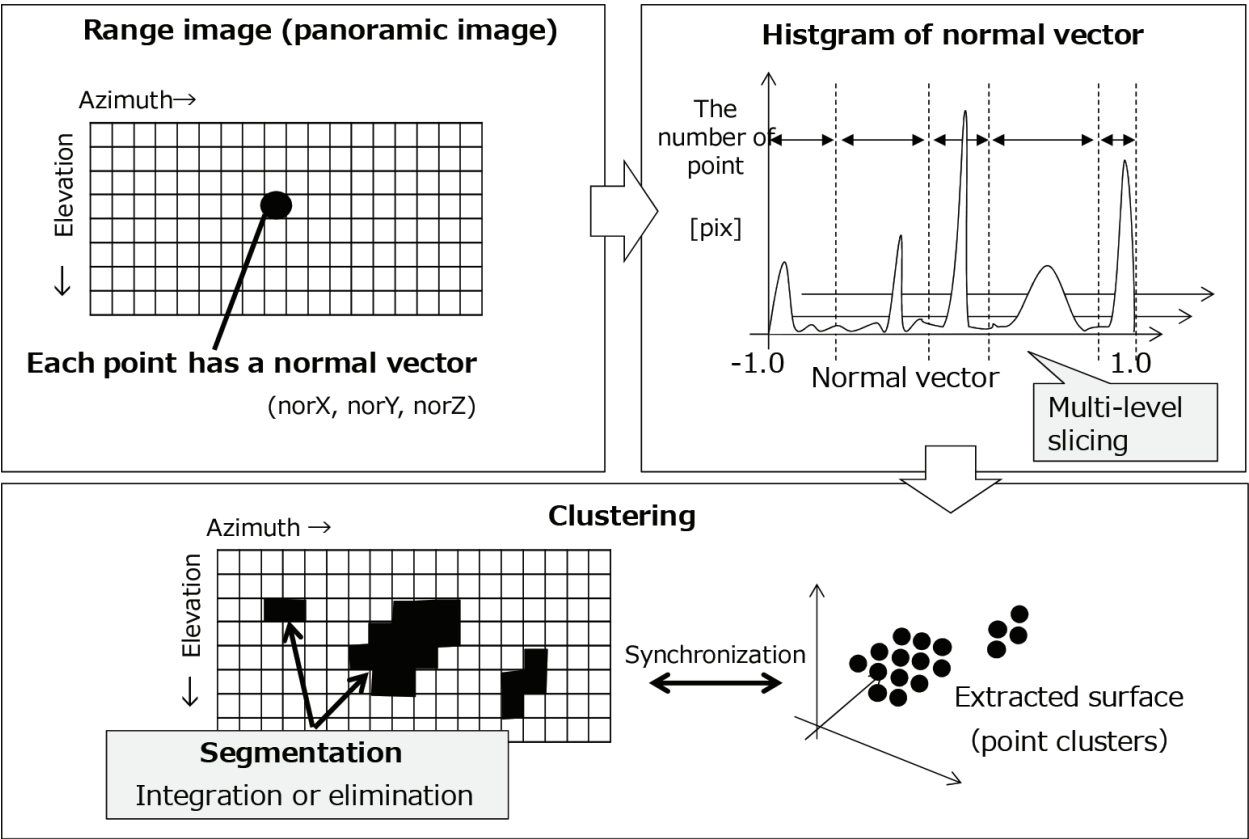


Figure 8. Normal vector-based point clustering.

planes to be integrated into a larger plane or deleted as a small segment. When a specified plane is extracted, the direction of a normal vector and the cluster number are available as initial value inputs. The point-cloud clustering methodology for extracting the intersection of planes as ridge lines requires appropriate initial values such as curvature, fitting accuracy and distances to closed points [19]. However, our approach can extract boundaries from a point cloud without these parameters.

### 3. Experiments

We conducted three experiments using point-cloud data acquired in indoor and outdoor environments. Here we present the point-cloud clustering results of these experiments.

#### 3.1. Experiment 1: indoor MMSs data

We selected a floor in our university as the indoor environment. We prepared a point cloud taken from an indoor MMSs (TIMMS, Nikon-Trimble), which consisted of a laser scanner, an omni-directional camera, inertial measurement units (IMU), and a wheel encoder, as shown in **Figure 9**. Acquired point-cloud data, point-cloud rendering results,



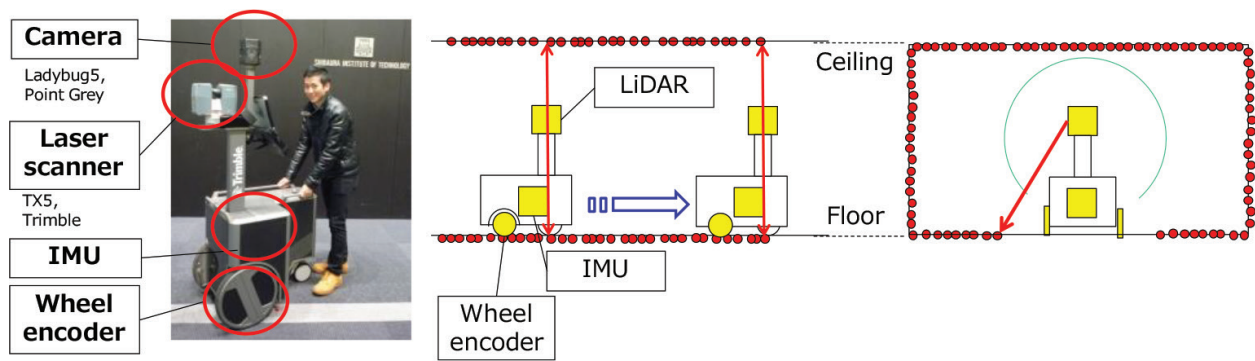


Figure 9. Indoor MMSs.

and point-cloud clustering results are shown in **Figure 10**. The results show that building features such as ceilings, beams, window shades, pillars, benches, and floors are classified clearly.

### 3.2. Experiment 2: terrestrial laser scanner data (1)

We selected a bridge as a study area in the outdoor environment. We acquired 25.9 million points using a terrestrial laser scanner (GLS-2000, TOPCON) from four viewpoints. Rendered point cloud, depth range image, and point-cloud clustering results are shown in **Figure 11**. The results show that vertical planes, horizontal planes, and natural features are classified clearly. The processing time for the clustering was 6.1 s (Intel Core i7-6567U 3.30 GHz, MATLAB).

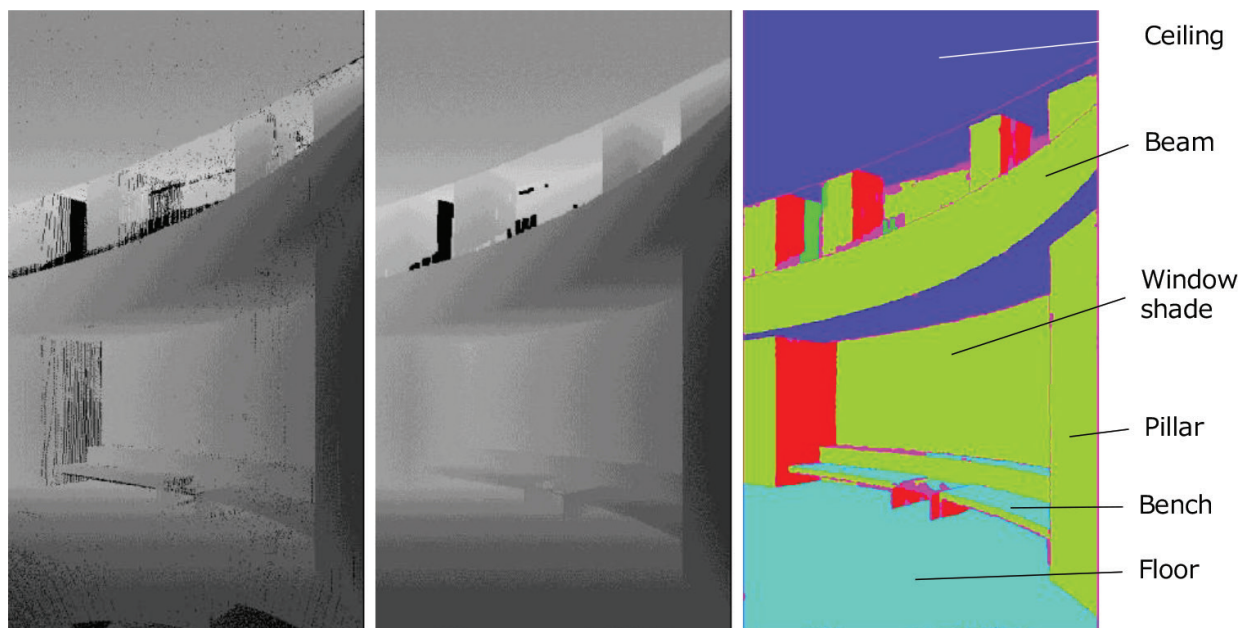
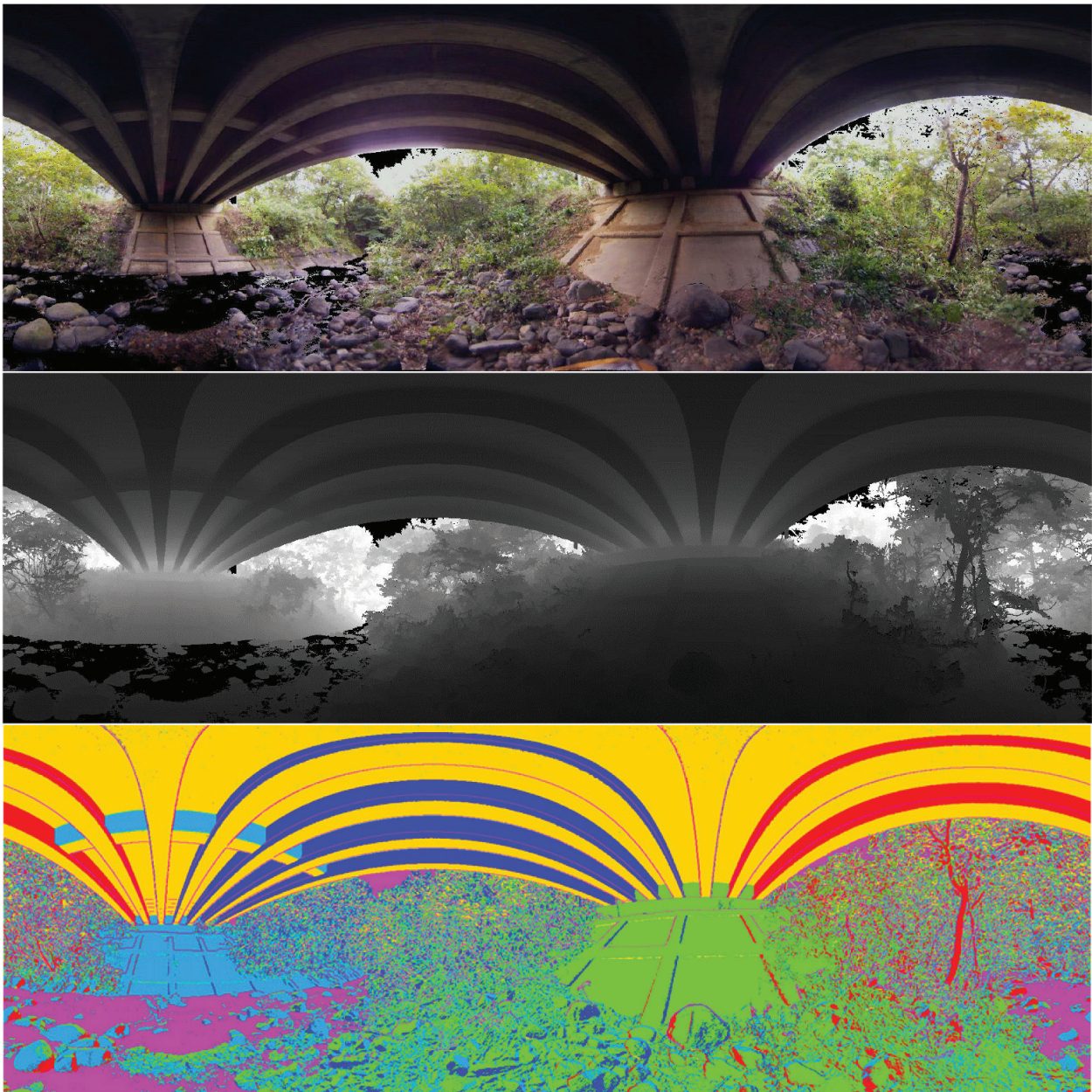


Figure 10. Point-cloud clustering result: rendered point cloud (left image), filtered point cloud (center image), and clustered point cloud (right image).



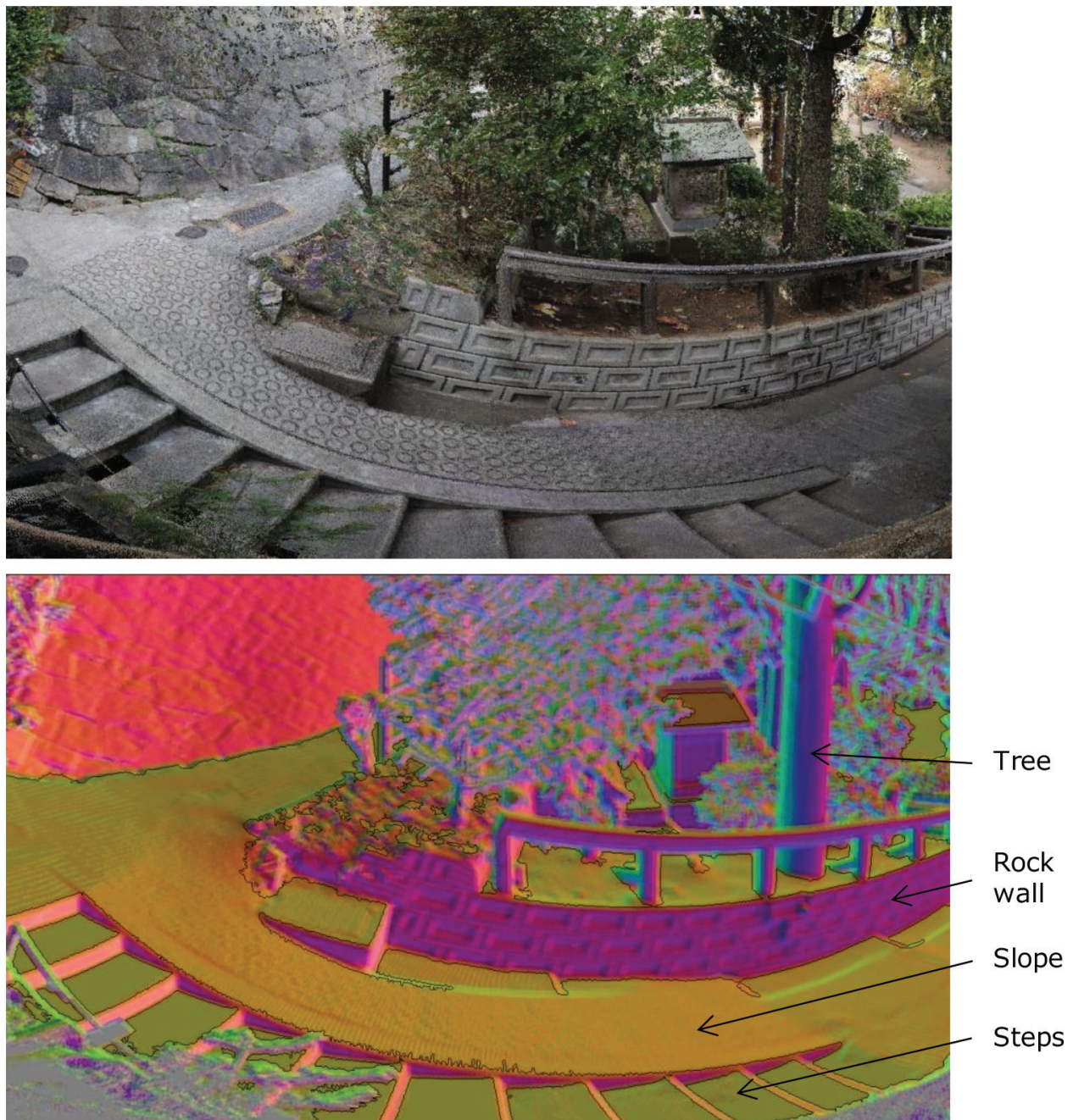


**Figure 11.** Point-cloud clustering result: colored point cloud (upper image), depth image (center image), and clustered point cloud (bottom image).

**3.3. Experiment 3: terrestrial laser scanner data (2)**

We selected a long narrow slope including slopes and stone steps as our study field. We prepared a point cloud acquired from 18 viewpoints over a wide area using a terrestrial laser scanner (VZ-400, RIEGL). Acquired point cloud and point-cloud clustering results are shown in **Figure 12**. The results show that features, such as steps, slopes, rock walls, and trees, are classified clearly. The processing time for the clustering was 11.4 s (Intel Core i7 2.80 GHz, MATLAB).





**Figure 12.** Point-cloud clustering result: colored point cloud (upper image) and clustered point cloud (bottom image).

## 4. Discussion

Our described processing flow in **Figure 2** can be extended from point-cloud clustering to polygon extraction [20], as shown in **Figure 13**.

After the “Normal vector classification in projected image,” when a small region has a similar direction with the neighboring region, the small region is merged into the neighboring

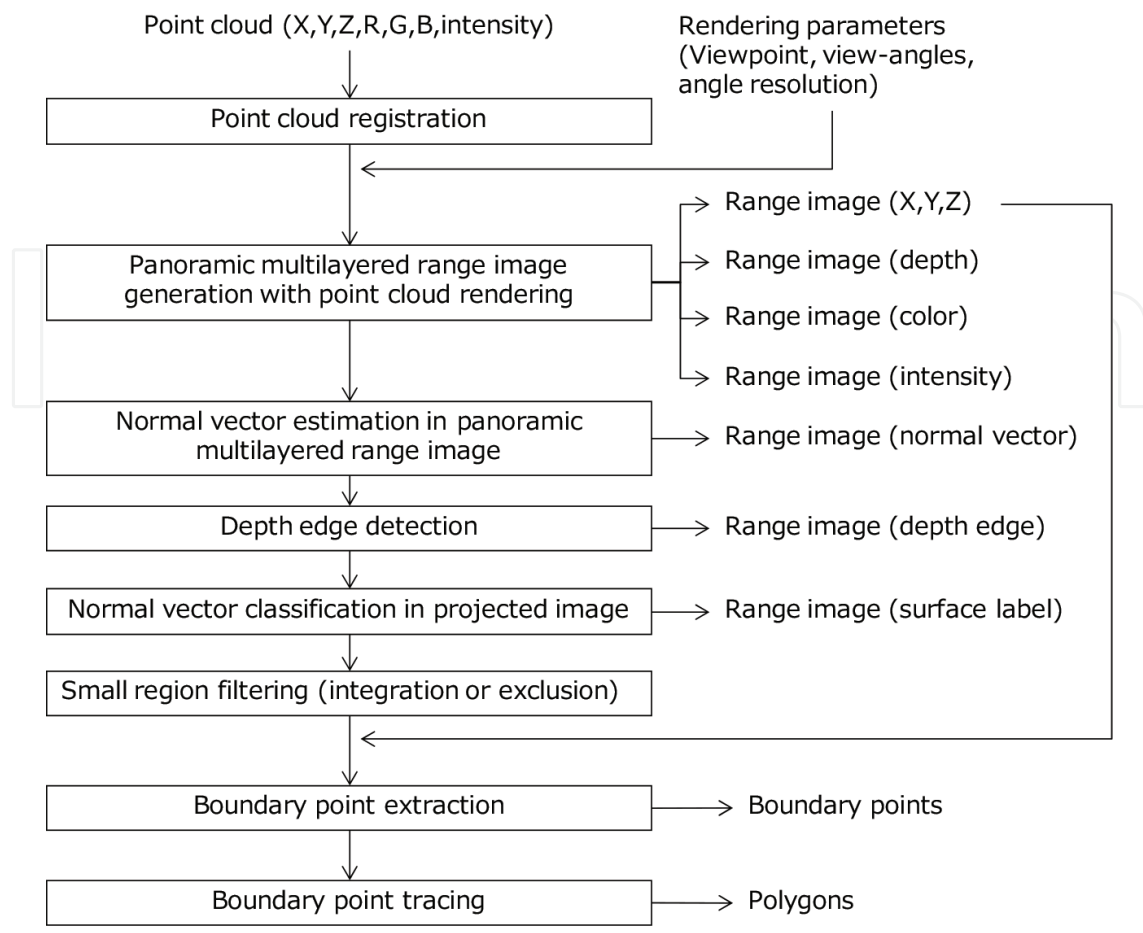
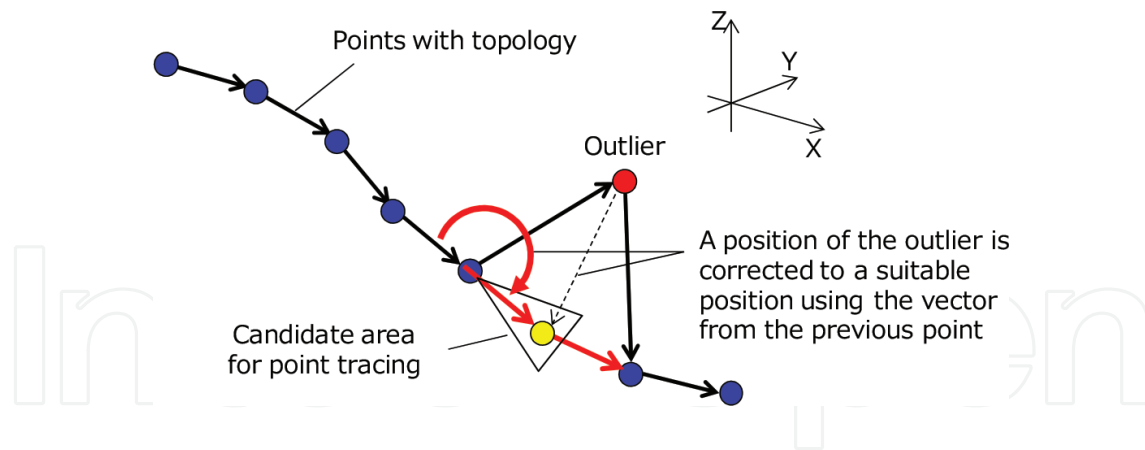


Figure 13. Overall processing flow.

region. Otherwise, the small region is deleted from the clustering result. Boundary points are extracted through “Boundary point extraction” from the clustering result. Moreover, polygons are extracted through “Boundary point tracing.” Boundaries of features can be extracted from the refined surfaces in a range image. Moreover, 3D polygons can be extracted with topology estimation using these extracted boundaries in the range image. In this procedure, point tracing connects points in the 3D space along the boundary, as shown in **Figure 14**. Although least squares fitting and polynomial fitting are generally applied for straight and curved line extraction from points, these fitting approaches require a straight-line recognition or curved-line recognition. When the point clouds include noises, RANSAC is a suitable approach for feature estimation. However, the RANSAC also requires the fitting procedure. Thus, tracing based on the region growing is applied to complex geometry extraction, as follows. First, a topology of points is estimated in a range image. Continuous 3D points can be extracted when a polyline or polygon is drawn in a range image. Next, a seed point is selected from the continuous 3D points for point tracing. Then, a possible next point is searched within a candidate area. The candidate area is determined using a 3D vector from the seed point. When a point exists within the candidate area, it is connected to the seed point. Otherwise, the point is assumed to be an outlier. A position of the outlier is corrected to a suitable position using the 3D vector from the seed point. Then, the connected point is assumed as the next

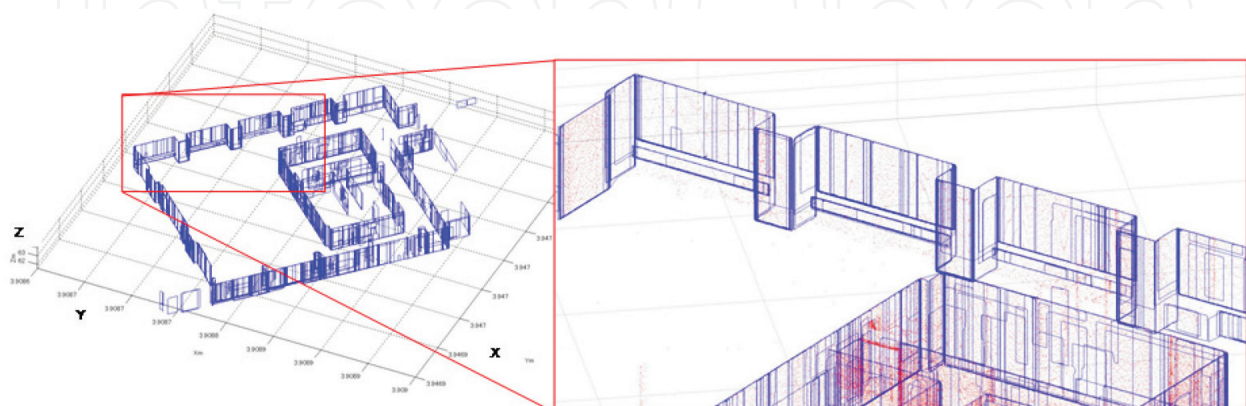


**Figure 14.** Point tracing.

seed point. These steps are iterated to close the geometry for the 3D smooth polygon generation. These procedures are applied to each rendered point cloud from arbitrary viewpoints.

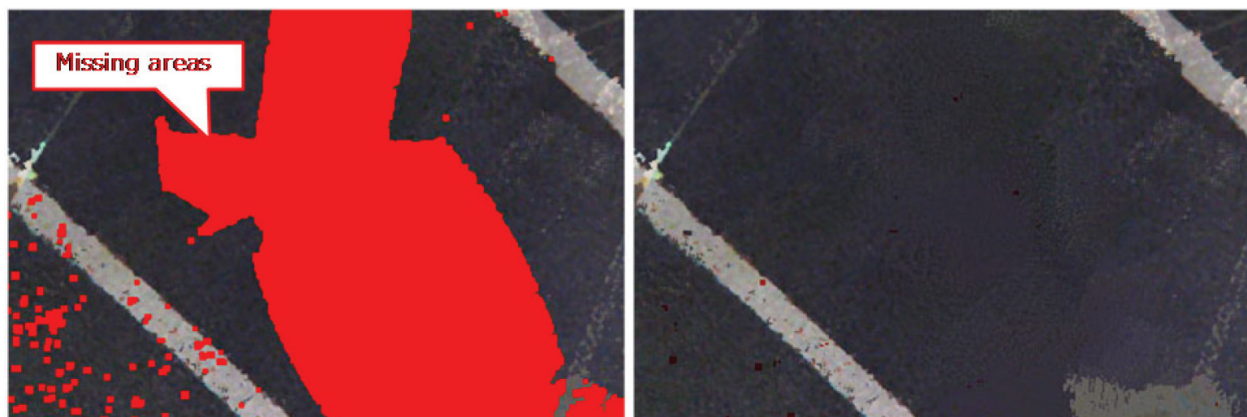
In indoor navigation and BIM, terrestrial laser scanners and indoor MMSs are used for colored point-cloud acquisition to generate floor maps and 3D data in an indoor environment. **Figure 15** shows the result of polygon extraction from the point cloud used in the first experiment.

However, missing areas and a non-uniform density of the point cloud would exist in point-cloud acquisition. This issue causes transparent and near-far effects in point-cloud visualization. To avoid these effects, we have developed a spatial interpolation based on point-based rendering in the point-cloud visualization and modeling. Nevertheless, when large missing and occluded areas exist, the spatial interpolation approach is inefficient and ineffectual. Therefore, we focused on a randomized algorithm to quickly find approximate nearest-neighbor matches between image patches for image inpainting [21]. The image inpainting aims to improve image quality with deletion works of scratches and unnecessary objects in an image and reconstruction works of the natural image. That is, scratches and unnecessary objects are replaced by other textures in the image [22], as shown in **Figure 16**. In manual works, these objects are replaced using image retouching software such as Adobe Photoshop. The inpainting approach is an automated procedure for image retouches.



**Figure 15.** Polygon extraction result.





**Figure 16.** Inpainted result: rendered point cloud (left image) and inpainted point cloud (right image).

## 5. Conclusion

We have focused on improving region-based point-cloud clustering in 3D modeling after point-cloud integration. We have also focused on region-based point clustering to extract a polygon from a massive point cloud, because it is difficult to estimate accurate edges from point clouds acquired with a laser scanner. First, we proposed a point-cloud clustering methodology on a panoramic layered range image generated from a massive point cloud with point-based rendering. Next, we conducted three experiments using laser scanning data to verify our methodology. The first experiment was 3D edge and surface extraction for indoor modeling using an indoor MMS. The second experiment was 3D edge and surface extraction for 3D bridge modeling using a terrestrial laser scanner. The third experiment was 3D edge and surface extraction for ground-surface and feature extraction using a terrestrial laser scanner. The results of these experiments confirm that our proposed methodology can achieve point-cloud clustering to extract features such as flat surfaces, slopes, and steps from complex environments in practical processing times.

## Acknowledgements

This work is supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications, Japan. This work is also supported by RIEGL Japan Co. Ltd., Nikon-Trimble Co. Ltd., and Sumire survey Co., Ltd. for data acquisition.

## Author details

Masafumi Nakagawa

Address all correspondence to: [mnaka@shibaura-it.ac.jp](mailto:mnaka@shibaura-it.ac.jp)

Shibaura Institute of Technology, Tokyo, Japan

## References

- [1] Boyko A, Funkhouser T. Extracting roads from dense point clouds in large scale urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2011;**66**(2011):S2-S12
- [2] Jiang X, Bunke H. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding*. 1999;**73**(2):183-199
- [3] Vosselman G, Gorte BGH, Sithole G, Rabbani T. Recognising structure in laser scanning point clouds. In: *ISPRS 2004: Proceedings of the ISPRS Working Group VIII/2: Laser Scanning for Forest and Landscape Assessment*; 2004. pp. 33-38
- [4] Tsai A, Hsu C, Hong I, Liu W. Plane and boundary extraction from LiDAR data using clustering and convex hull projection. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2010;**XXXVIII**(Part 3A):175-179
- [5] Pu S, Vosselman G. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2009;**64**(6):575-584
- [6] Zhou Q, Neumann U. Fast and extensible building modeling from airborne LiDAR data. In: *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)*. 2008. p. 8
- [7] Denis E, Burck R, Baillard C. Towards road modelling from terrestrial laser points. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2010;**XXXVIII**(Part 3A):293-298
- [8] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*. 2007;**26**(2):214-226
- [9] Shade J, Gortler S, He L, Szeliski R. Layered depth images. In: *SIGGRAPH '98*. 1998. pp. 231-242
- [10] Verbree E, Zlatanova S, Dijkman S. Distance-Value-Added Panoramic Images as the Base Data Model for 3D-GIS. *Panoramic Photogrammetry Workshop*. 2005
- [11] Linsen L, Müller K, Rosenthal P. Splat-based ray tracing of point clouds. *Journal of WSCG*. 2007;**15**(1-3):51-58
- [12] Nakagawa M. Point cloud clustering for 3D modeling assistance using a panoramic layered range image. *Journal of Remote Sensing Technology*. 2013;**1**(3):10
- [13] Chevallier N, Maillot Y. Boundary of a non-uniform point cloud for reconstruction. In: *SoCG '11 Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*. 2011. pp. 510-518
- [14] Fabio R. From point cloud to surface: The modeling and visualization problem. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2003;**XXXIV-5/W10**

- [15] Böhm J, Pateraki M. From point samples to surfaces, on meshing and alternatives. *ISPRS Image Engineering and Vision Metrology*. 2006;**XXXVI**:50-55
- [16] Golovinskiy A, Funkhouser T. Min-cut based segmentation of point clouds. In: *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*; 2009. p. 6
- [17] Shapovalov R, Velizhev A. Cutting-plane training of non-associative Markov network for 3D point cloud segmentation. In: *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*; 2011. p. 8
- [18] Biosca M, Luis Lerma J. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2008;**63**(1):84-98
- [19] Kitamura K, D'Apuzzo N, Kochi N, Kaneko S. Automated extraction of break lines in tils data of real environment. *International Archives of Photogrammetry and Remote Sensing*. 2010;**38**(5):331-336
- [20] Nakagawa M, Kataoka K, Yamamoto T, Shiozaki M, Ohhashi T. Panoramic rendering-based polygon extraction from indoor mobile LiDAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2014;**XL**(4):181-186
- [21] Barnes C, Shechtman E, Finkelstein A, Dan Goldman B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 2009;**28**(3)
- [22] Nakagawa M, Tanaka S, Yamamoto T. Colorerd point cloud reconstruction based on image inpainting. In: *The 36th Asian Conference on Remote Sensing 2015 (ACRS2015)*; 2015. p. 7