A POINT-BASED RENDERING FOR 3D POLYGON EXTRACTION IN INDOOR ENVIRONMENT

Masafumi Nakagawa[1], Tatsuya Yamamoto[1], Konosuke Kataoka[1], Makoto Shiozaki[2], Tetsuya Ohhashi[2]
[1]Shibaura Institute of Technology, 3-7-5, Toyosu, Koto-ku, Tokyo 135-8548 Japan
Email: mnaka@shibaura-it.ac.jp
[2]Nikon-Trimble Co., Ltd., 2-16-2, Minamikamata, Ota-ku, Tokyo 144-0035, Japan
Email: shiozaki.makoto@nikon-trimble.net

KEY WORDS: Point-cloud, Point-based rendering, Indoor mobile mapping system, Polygon extraction, Indoor GIS

ABSTRACT: Point-cloud clustering is an essential technique for modeling massive point clouds. We focus on the region-based point clustering to extract a polygon from a massive point cloud. In region-based clustering, Random Sample Consensus (RANSAC) is a suitable approach for estimating surfaces. However, local workspace selection is required to improve a performance in a surface estimation from a massive point cloud. Moreover, with conventional RANSAC, it is hard to determine whether a point lies inside or outside a surface. In this paper, we propose a method for panoramic rendering-based polygon extraction from indoor mobile LiDAR data. Next, we confirm that our proposed methodology can achieve polygon extraction through point-cloud clustering from an indoor environment.

1. INTRODUCTION

Point-cloud clustering is an essential technique for modeling massive point clouds acquired with a terrestrial laser scanner or mobile laser scanner. There are three clustering approaches in point-cloud clustering: model-based clustering (Boyko et al. 2011), edge-based clustering (Jiang et al. 1999), and region-based clustering (Vosselman et al. 2004). Edge-based and region-based clustering are often used to model unknown objects (Tsai et al. 2010). These approaches also focus on geometrical knowledge (Pu, et al. 2009) and 2D geometrical restrictions, such as the depth from a platform (Zhou, et al. 2008) and discontinuous point extraction on each scanning plane from the mobile mapping system (Denis et al. 2010) to extract simple boundaries and features in urban areas.
Point-cloud data acquired in urban areas and indoor environments often include many complex features with unclear boundaries. Thus, we focus on the region-based point clustering to extract a polygon from a massive point cloud. In region-based clustering, Random Sample Consensus (RANSAC) (Schnabel et al. 2007) is a suitable approach for estimating surfaces. However, local workspace selection is required to improve a performance in a surface estimation from a massive point cloud. Moreover, with conventional RANSAC, it is hard to determine whether a point lies inside or outside a surface.
In this paper, we propose a method for panoramic rendering-based polygon extraction from indoor mobile LiDAR data. First, we propose a point-cloud clustering methodology for polygon extraction on a panoramic range image generated with point-based rendering from a massive point cloud. Next, we describe an experiment that was conducted to verify our methodology with an indoor mobile mapping system. Finally, we confirm that our proposed methodology can achieve polygon extraction through point-cloud clustering from a complex indoor environment.

2. METHODOLOGY

Figure 1 shows our proposed methodology. It consists of: (1) viewpoint decision for point-based rendering; (2) point-based rendering; (3) normal vector clustering for surface estimation; (4) point-cloud interpolation using a rectangular template; and (5) point tracing.

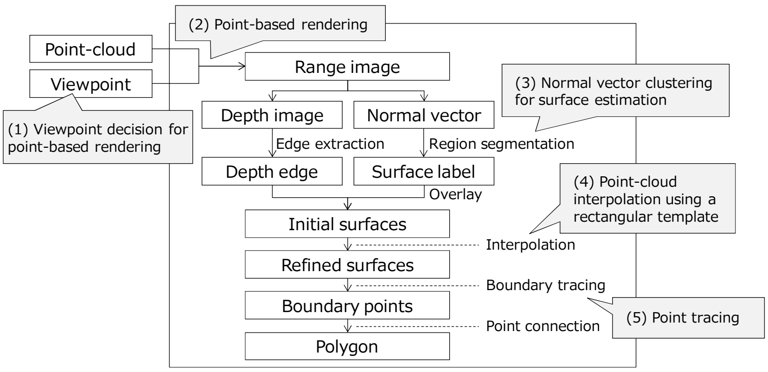

Figure 1. The five components of processing flow

2.1 Viewpoint decision for point-based rendering

Viewpoints are selected in the random point cloud for point-based rendering. The viewpoints are selected in point-cloud data through two steps. In the first step, an orthobinary image is generated from the point cloud to represent a rough floor surface as a viewpoint candidate. In the next step, the orthoimage is eroded with morphology processing to generate a viewpoint candidate network. Intersections on the network are selected as the viewpoints for point-based rendering.

2.2 Point-based rendering

Point-cloud visualization has two issues. The first is the near-far problem caused by distance differences between the viewpoint and the scanned points. The second is the transparency effect caused by rendering hidden points among near-side points. These effects degrade the quality of a point-cloud visualization. Splat-based ray tracing (Linsen et al. 2007) is a methodology that generates a photorealistic curved surface on a panoramic view using normal vectors from point-cloud data. The curved-surface description is inefficient in representing urban and natural objects as Geographical Information System data. Thus, we have applied a point-based rendering application with a simpler filtering algorithm (Nakagawa 2013) to generate panoramic range images from a random-point cloud. The processing flow of point-based rendering is described in Figure 2.

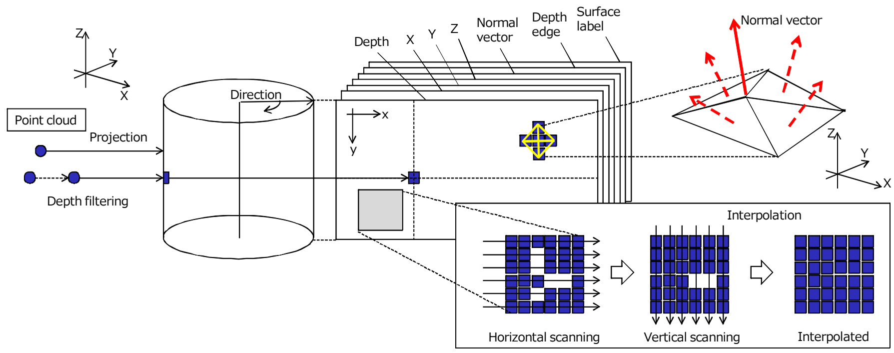

Figure 2. Point-based rendering

First, the point cloud is projected from 3D space to panorama space. This transformation simplifies viewpoint translation, filtering, and point-cloud browsing. The panorama space can be represented by a spherical, hemispherical, cylindrical, or cubic model. Here, the cylindrical model is described for wall modeling. The measured point data are projected onto a cylindrical surface, and can be represented as range data. The range data can preserve measured point data such as a depth, X, Y, Z, and some processed data in the panorama space in a multilayer style.
Second, the generated range image is filtered to generate missing points in the rendered result using distance values between the viewpoint and objects. Two types of filtering are performed in the point-based rendering. The first is a depth filtering with the overwriting of occluded points. The second is the generation of new points in the no-data spaces in the range image. New points are generated with the point tracking filter developed in this study.
Moreover, a normal vector from each point is estimated in the range image. First, a point and its neighbors in the range image are selected. Second, triangulation is applied to these points as vertexes to generate faces. Then, the normal vector on each triangle is estimated using 3D coordinate values of each point. In this research, an average value of each normal vector is used as the normal vector of a point. These procedures are iterated to estimate the normal vectors of all points.

2.3 Normal vector clustering for surface estimation

Normal vectors of all points are grouped to detect regions in a range image as a point-cloud classification. We applied multilevel slicing as a simple algorithm to classify normal vectors. The accuracy of point-cloud classification can be improved with several approaches such as the Mincut, Markov network-based, and fuzzy-based algorithms. Moreover, building knowledge is used as a restriction in the normal vector and point-cloud classification. In general, walls in a room and building consist of parallel and orthogonal planes. Thus, four clusters in a horizontal direction are enough to detect walls in a general indoor environment. Although cylindrical surfaces are divided into some clusters, these surfaces can be reconstructed using surface merging. The processing flow of normal vector clustering with restrictions is described below. First, stronger peaks are extracted from a histogram of normal vectors. More than one strong peak is required to detect seed points in each approximate 90° change in horizontal direction. Next, boundaries of clusters are generated from the peaks of the histograms. Then, the normal vectors and point clouds are grouped into four clusters. Finally, initial 3D surfaces are estimated from the grouped normal vectors and point cloud.

2.4 Point-cloud interpolation with a rectangular template

Estimated 3D initial surfaces are refined in a point-cloud interpolation procedure. When flat and cylindrical surfaces are projected into a range image based on a cylindrical model, these surfaces are represented as rectangles with the following two restrictions. The first restriction is that points have the same X- and Y-coordinate values along the y-direction in the range image. The second restriction is that the points have the same Z-coordinate values along the x-direction in the range image. Based on these restrictions, point interpolation is applied along the x- and y-directions in the range image, as shown in Figure 3. The point interpolation is as follows. First, a rectangular template is fitted to projected points in a range image. Next, missing points are detected in the rectangular template. Finally, the missing points are interpolated using neighboring points. When other objects exist in a rectangular template, the overlapped area is excluded from point interpolation.

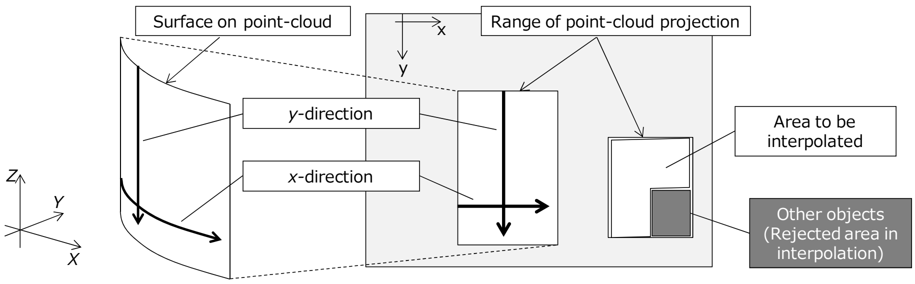

Figure 3. Point-cloud interpolation with a rectangular template in a range image

2.5 Point tracing

Boundaries of features can be estimated from the refined surfaces in a range. Moreover, 3D polygons can be extracted with topology estimation using these boundaries in the range image. In this procedure, a point tracing is required to connect points in 3D space along the boundary, as shown in Figure 4. In general, least squares fitting and polynomial fitting are applied to extract straight and curved lines from points. However, these approaches require a decision whether straight lines or curved lines are to be extracted before the fitting procedure. In this paper, we wish to extract polygons with a combination of straight and curved lines. Thus, we propose point tracing based on the region-growing approach to extract complex geometry as follows. First, a topology of points is estimated in a range image. Next, a position for the next point is checked after a seed-point selection. In this step, the position is checked to find whether a possible next point exists or not within a candidate area for point tracing. These steps are then iterated until the geometry is closed. Finally, 3D points are connected to represent a smooth 3D polygon.

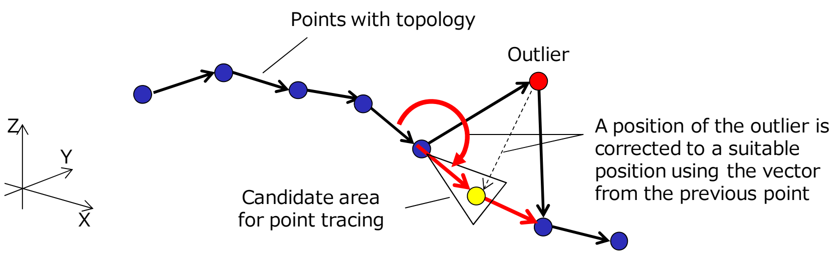

Figure 4. Point tracing

3. EXPERIMENT

An entrance foyer consisting of a large room (8.72 m × 54.00 m width × 4.10 m height) in our university was selected as our study area (see Figure 5). The study area consisted of flat and cylindrical walls, square and cylindrical pillars, a grilled ceiling, doors with glass, and windows. These objects were representative flat and cylindrical surfaces. In the experiment, we used the Trimble Indoor Mobile Mapping System (TIMMS) integrated with an Inertial Measurement Unit (POS LV, Applanix), a wheel encoder, a LiDAR system (TX5, Trimble), and an omnidirectional camera (Ladybug 5, Point Grey) (see Figure 6). We acquired a 660-million color point cloud with TIMMS (see Figure 7).

Figure 5. Study area

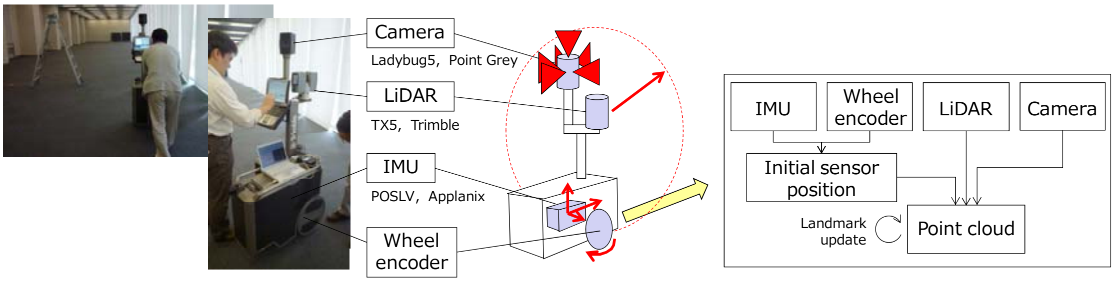

Figure 6. TIMMS

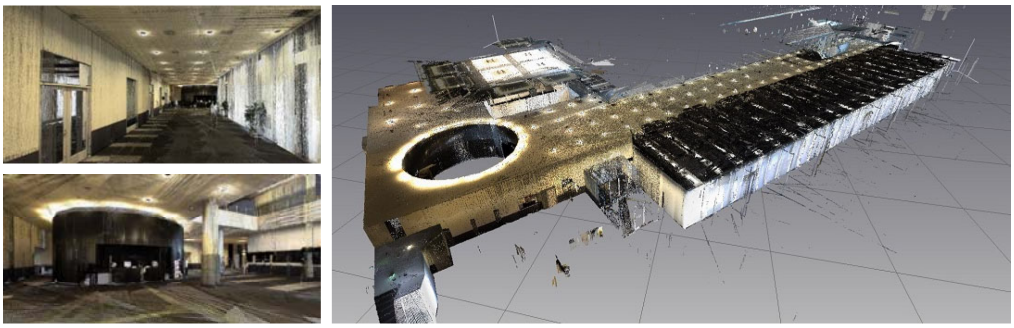

Figure 7. Acquired colored point cloud

In our experiment, 64 points were extracted as viewpoint candidates for point-based rendering, as shown in Figure 8. The point cloud taken from TIMMS was rendered from these viewpoints.

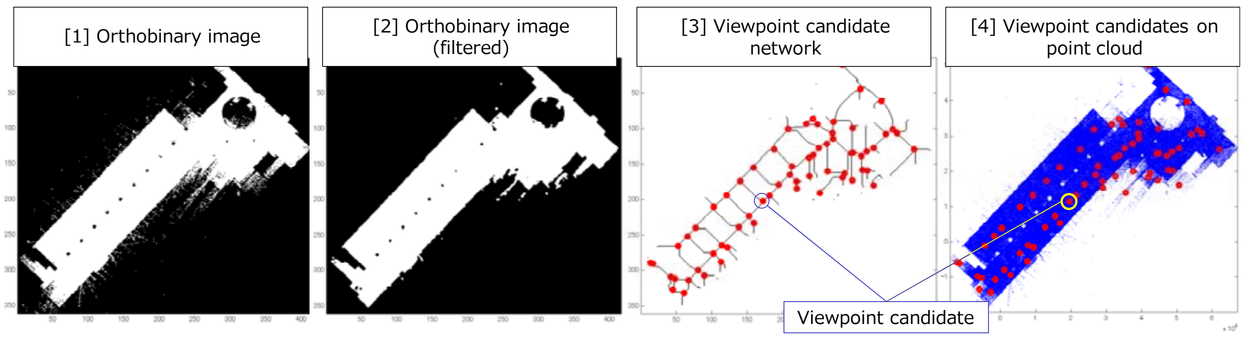

Figure 8. Viewpoint candidates

Figure 9 shows results after point-based rendering and point clustering from a viewpoint. Figure 9 includes a depth image, a filtered depth image, normal vectors, labeled surfaces, and initial surfaces (overlay of depth edge and labeled surfaces). Each vertical axis shows height direction and each horizontal axis shows direction. Intensity values in the depth image and filtered depth image indicate the depth from the viewpoint. Moreover, intensity values in the normal vectors and labeled surfaces indicate the horizontal direction of the point cloud. In addition, color values in the initial surfaces indicate labels of surfaces. In this experiment, spatial resolution was set as 0.2° in the horizontal direction and 2 cm in the height direction.

Figure 10 shows a rendered point cloud from a viewpoint in 3D space. The left image shows the input point cloud and the right image shows a result after polygon extraction. Processing time for the panoramic image conversion and polygon extraction was several minutes in total for each viewpoint using an Intel core i7 2.80 GHz processor with MATLAB (single thread).

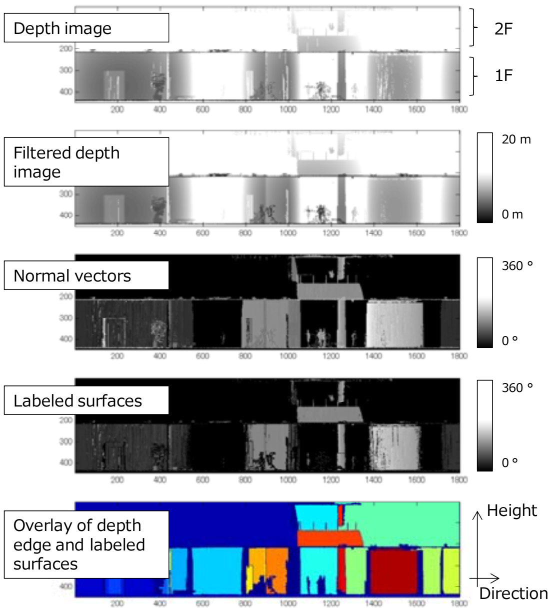

Figure 9. Results after point-based rendering and point clustering

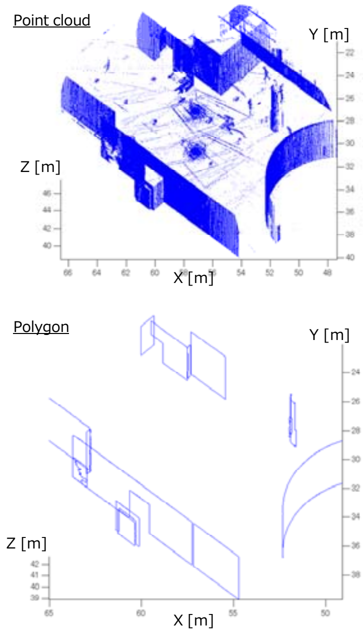

Figure 10. Point cloud (top image) and polygon extraction result (bottom image)

4. DISCUSSION

Parts of the results of polygon extraction from the point cloud are shown in Figure 11. This figure includes examples of general building features, such as a flat wall, a door, and a cylindrical wall. Each row shows a result of point-cloud visualization and extracted boundaries. We have confirmed that point-cloud interpolation in a range image achieved spike noise filtering and geometry smoothing. Moreover, we have confirmed that noise such as the pedestrian was also successfully filtered from the point cloud.

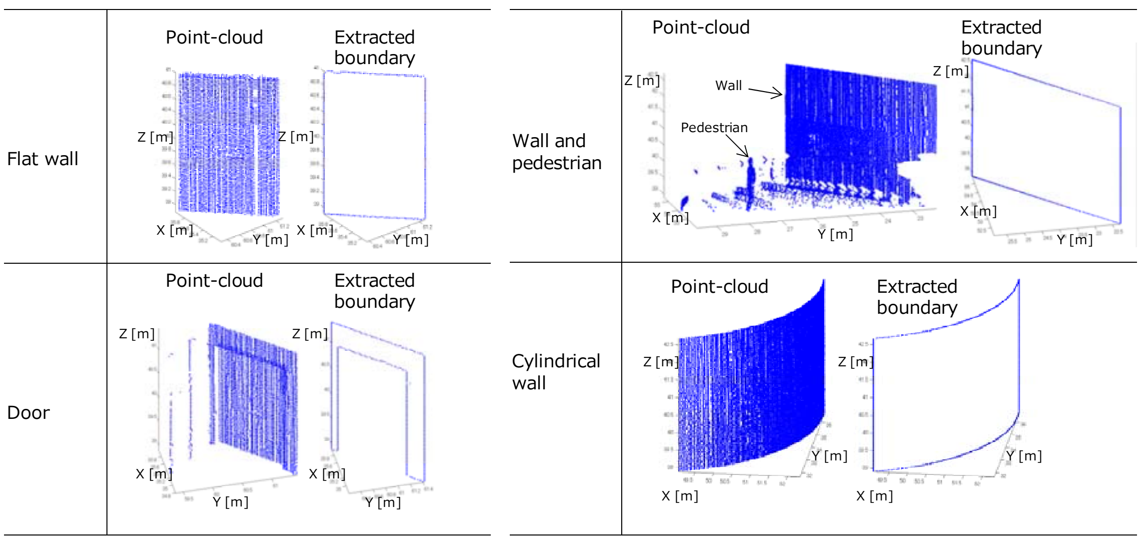

Figure 11. Parts of results of polygon extraction from point cloud

The left image in Figure 12 shows integrated results for polygon extraction from 64 viewpoints. Our approach extracted 892 polygons from the point cloud fully automatically. We also conducted manual editing to evaluate the performance of polygon extraction. The right image in Figure 12 shows the result after editing. We confirmed that 863 polygons were extracted from the point cloud successfully. We deleted 29 polygons as failures in the polygon extraction. Thus, the success rate of polygon extraction was 97% (863/892) in this experiment. As shown in the left image in Figure 12, some polygons that were extracted were failures. Our investigation showed that these failures were caused by LiDAR measurement noise, such as light reflection errors and moving object measurement. Although noise was almost eliminated, the remained noise in the range image affected the point-cloud interpolation.

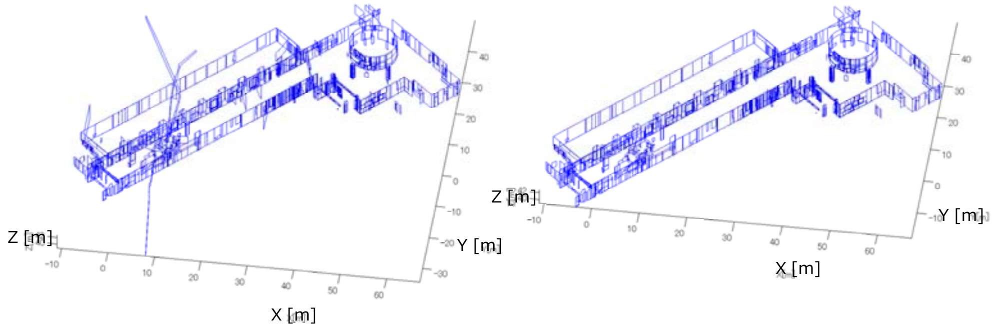

Figure 12. Integrated results in polygon extraction from 64 viewpoints (left image) and the result after manual editing (right image)

5. SUMMARY

We have proposed a method for panoramic rendering-based polygon extraction from indoor mobile LiDAR data. Our aim was to improve region-based point cloud cluster modeling after point-cloud registration. Our proposed methodology consisted of the viewpoint decision for point-based rendering, the point-based rendering, the normal vector clustering for surface estimation, the point-cloud interpolation with a rectangular template, and point tracing. Next, we described an experiment that was conducted to verify our methodology with an indoor mobile mapping system (TIMMS) in an indoor environment that included flat and cylindrical surfaces. In this experiment, we extracted wall -surfaces using a rendered point cloud from 64 viewpoints over a wide indoor area. Finally, we confirmed that our proposed methodology could achieve polygon extraction through point-cloud clustering from a complex indoor environment.

REFERENCES

[1] Boyko, A., Funkhouser, T., 2011, Extracting roads from dense point clouds in large scale urban environment, ISPRS Journal of Photogrammetry and Remote Sensing, 66 (2011) S2–S12.

[2] Jiang, X., Bunke, H., 1999, Edge Detection in Range Images Based on Scan Line Approximation, Computer Vision and Image Understanding, Vol.73, No.2, pp.183-199.

[3] Vosselman, G., G. H. Gorte, B., Sithole, G., Rabbani, T., 2004, 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, pp. 33-38.

[4] Tsai, A., Hsu, C., Hong, I., Liu, W., 2010, Plane and Boundary Extraction from LiDAR data using Clustering and Convex Hull Projection, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII - Part 3A, pp.175-179.

[5] Pu, S., Vosselman, G., 2009, Knowledge based reconstruction of building models from terrestrial laser scanning data, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 64, Issue 6, pp. 575–584.

[6] Zhou, Q., Neumann, U. 2008, Fast and Extensible Building Modeling from Airborne LiDAR Data, ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), pp.8.

[7] Denis, E., Burck, R., Baillard, C. 2010, Towards road modelling from terrestrial laser points, IAPRS, Vol. XXXVIII, Part 3A, pp.293-298.

[8] Schnabel, R., Wahl, R., Klein, R., 2007, Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum, 26(2), 214-226.

[9] Linsen, L., M¨uller, K., Rosenthal, P., 2007, Splat-based Ray Tracing of Point Clouds, Journal of WSCG, Vol.15, Issue: 1-3, pp.51-58.

[10] Nakagawa, M., 2013, Point Cloud Clustering for 3D Modeling Assistance Using a Panoramic Layered Range Image, Journal of Remote Sensing Technology, Vol.1, Iss.3, 10.pp.