Reconstruction of Polygonal Prisms from Point-Clouds of Engineering Facilities

Akisato Chida and Hiroshi Masuda[*]
Department of Information Science and Engineering, the University of Electro-Communications, Tokyo, Japan

Abstract

The advent of high-performance terrestrial laser scanners has made it possible to capture dense point-clouds of engineering facilities. 3D shape acquisition from engineering facilities is useful for supporting maintenance and repair tasks. In this paper, we discuss methods to reconstruct box shapes and polygonal prisms from large-scale point-clouds. Since many faces may be partly occluded by other objects in engineering plants, we estimate possible box shapes and polygonal prisms and verify their compatibility with measured point-clouds. We evaluate our method using actual point-clouds of engineering plants.

Key words: geometric modeling, point processing, shape reconstruction, terrestrial laser scanner

1. Introduction

3D shape acquisition from engineering facilities is very important for maintenance and repair tasks. It is well known that model-based planning can intensively reduce the rework of maintenance and repair tasks for engineering facilities. Many CAD systems provide capabilities of task simulation based on 3D solid models. However, reliable 3D models of old engineering facilities rarely exist, because they were typically built one or more decades ago based on 2D drawings and have been repeatedly renovated in their long lifecycles. In most cases, it is required that 3D models are newly created based on existing facilities.

The advent of high-performance terrestrial laser scanners has made it possible to capture dense point-clouds of engineering facilities. The state-of-the-art phase-based laser scanners can capture about one million points in a second. By using such laser scanners, large engineering facilities can be represented with a huge number of discrete points. When a facility is measured at intervals of 6.3 mm at the distance of 10 m, the number of points is about fifty millions per each scan. An engineering facility is typically measured at dozens of places to reduce occlusions. Then the total number of points becomes hundreds of millions. Large-scale point-clouds represent faithful as-is shapes, but their data sizes are too large to handle with common PCs.

It is often required to convert large-scale point-clouds to more concise surface models. For shape reconstruction of engineering facilities, many shape reconstruction methods for pipe structures have been reported [1, 2, 4, 5, 8, 10]. In these researches, cylindrical surfaces are extracted from point-clouds, and then pipe structures are reconstructed by extending and connecting cylinders. On the other hand, the extraction of planar objects is more complicated. Although planar surfaces can be easily extracted using the RANSAC method or the region-growing method, the estimation of their boundaries is not a trivial problem.

In our previous work, we proposed a method for efficiently extracting planes from large-scale point-clouds [3]. In engineering facilities, we can observe that the main objects are mostly box-shaped or polygonal prisms except pipe structures. In this paper, we discuss methods to extract box shapes and polygonal prisms from large-scale point-clouds.

[*] Corresponding author email: h.masuda@uec.ac.jp

2. Overview

In engineering facilities, most objects consist of planes and cylinders. Cones and tori surfaces also appear, but they are mostly used to connect cylindrical pipes and their sizes and positions can be estimated based on cylinders [5]. Other popular objects in engineering facilities are cuboids and polygonal prisms. In this paper, we call cuboids as box shapes.

Fig. 1 illustrates our shape reconstruction method. An engineering facility is represented with point-clouds. In our system, multiple scan data are separately processed. We suppose that the coordinates of points in each scan are represented on the scanner-centered coordinate system. Then points can be ordered on the 2D image (Fig. 1(b)). In this paper, we call points that are ordered in a lattice manner as the 2D map. We mainly process a point-cloud on the 2D map.

Then planes are extracted from a point-cloud. In Fig. 1(c), planar regions are shown in different colors on the 2D map. Our plane detection method is based on our previous work [3]. Next, we estimate box shapes using the relationships among planar regions. Since points are discrete and noisy, the boundary of each planar region cannot be precisely obtained. In addition, faces are often partly occluded by other objects. Therefore, we extract possible pairs of planar regions to reconstruct box shapes (Fig. 1(d)). In addition, polygonal prisms are also estimated using planar regions (Fig. 1(e)). Estimated objects are confirmed using visibility check (Fig. 1(e)). If estimated objects are inconsistent with a point-cloud, they are discarded.

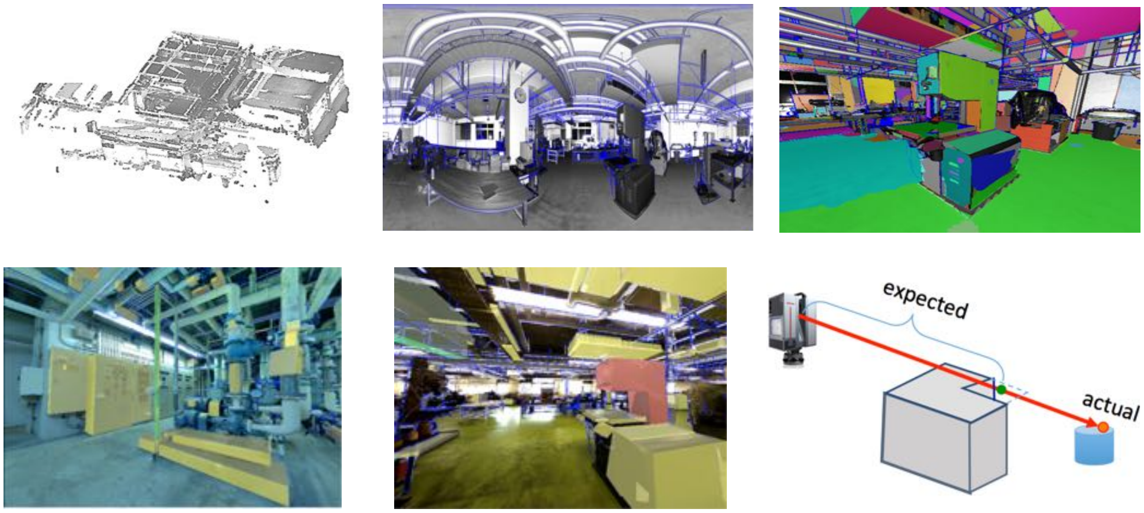

Fig. 1. Process of shape reconstruction. (a) point-cloud, (b) points on a 2D lattice, (c) extraction of planar regions, (d) estimation of box shapes, (e) estimation of polygonal prisms, and (f) confirmation using visibility check.

3. Extraction of Planes and Cylinders

3.1 Generation of 2D map

The terrestrial laser scanners emit laser pulses and measure the round-trip travel time of the laser pulses reflected from objects. Fig. 2(a) shows a typical mechanism of terrestrial laser scanners. The laser scanner continuously emits laser pulses from the light source. The directions of laser pulses are vertically moved by the spinning miller and horizontally moved by the rotating body of the laser scanner. The laser scanner stores the directions and the round-trip travel times, which are converted into 3D coordinates in the post-process phase.

According to this mechanism, points in a point-cloud file are regularly ordered. The directions of laser pulses can be represented using the azimuth angle θ and the zenith angle ϕ. When the sampling intervals of each angle are constant, points are regularly ordered in the angle space. Fig. 2(b) shows the 2D map that are generated from regularly ordered points. The brightness of each pixel in the image shows the strength of reflected laser pulses, which is typically output with coordinates from the laser scanner. One of popular formats for point-clouds is the PTX format, in which points are ordered in a 2D lattice manner. Since each point can be mapped onto (I, J) in the 2D map, neighbor points can be quickly obtained.

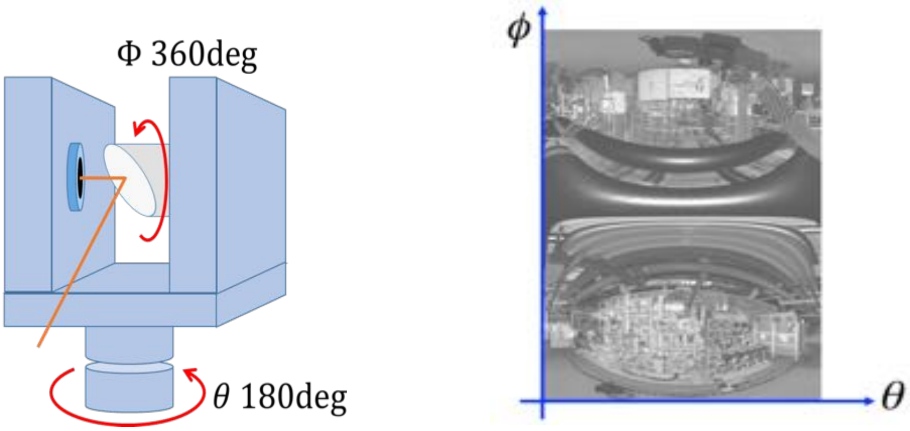

Fig. 2. Mechanism of laser scanners. (a) terrestrial laser scanner, and (b) a point-cloud on angle space.

3.2 Surface detection

The RANSAC method is often used to detect planar and cylindrical regions in a point-cloud [9]. In our system, planes and cylinders are extracted from point-clouds using our previously proposed methods [3]. We first estimate the normal vector at each point. Normal vectors are calculated by applying the principal component analysis to neighbor points on the 2D map.

To extract planes using the RANSAC method, three points are randomly selected and a plane equation is calculated. Then the number of points on the plane is counted. This process is iterated many times and the system maintains the plane equation with the maximum number of points. In the case of cylinder detection, two points are randomly selected and a cylinder is calculated using the coordinates and normal vectors [9].

It is well known that the RANSAC method is prohibitively time-consuming when it is applied to a large-scale point-cloud. To extract surfaces in a practical time, it is necessary to segment a point-cloud into small subsets.

In our method, we subdivide a point-cloud into continuous regions. When two adjacent points are on a continuous surface, their distance can be estimated as:

$$s = \frac{|\boldsymbol{p}|^2 \, \Delta\phi}{|(\boldsymbol{p}, \boldsymbol{n})|} \tag{1}$$

where $\boldsymbol{p}$ is a coordinate; $\boldsymbol{n}$ is the normal vector of the plane; $\Delta\phi$ is the interval of the azimuth and zenith angles. We calculate distances between point $\boldsymbol{p}$ and the four neighbors and describe the distances as d_1, d_2, d_3, and d_4. We regard that the two points are on a continuous surface only when

$$d_i < k \cdot s \quad (i = 1,2,3,4) \tag{2}$$

The variable k is a constant value. We specified k=3 in this paper.

We detect each continuous surface using the region-growing method. We select an arbitrary point as a seed and extract the continuous region according to the criteria (2). We repeat this process until all points are segmented into continuous regions. Fig. 3 shows an example of segmented regions. In this example, 1542 continuous regions are extracted. We discarded small regions that consist of less than 300 points.

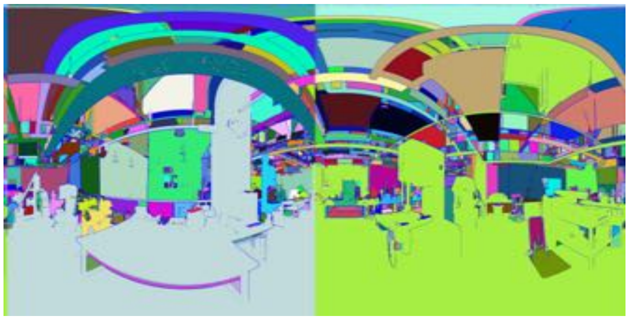

Fig. 3. Segmentation to continuous regions.

3.3 Detection of cylinders and planes

Planes are detected in each continuous region using the RANSAC method. We describe the total number of points in a point-cloud as N and the number of points on a certain surface as n. The calculation time of the RANSAC method is determined according to the ratio n/N. When the ratio is very small, the number of iterations required for the RANSAC method becomes prohibitively large.

Fortunately, large continuous regions in manufacturing facilities include large planar floors or walls. Since floor and wall planes are very large, they can be detected in a small number of iterations. When floors and walls are eliminated, each continuous region can be segmented into small continuous regions. In our method, continuous regions are recursively segmented each time when a surface is extracted and it is removed from the continuous region.

In our method, a planar region and a cylindrical region are simultaneously extracted from a continuous region, and the larger region is selected. This method avoids a cylindrical surface to be subdivided into a lot of strip-shaped planes. Surface detection is repeated until regions with more than m points cannot be extracted. In this paper, we set m to 300.

Fig. 4(a) shows planar surfaces and Fig. 4(b) shows cylindrical surfaces. Fig. 5 shows planar and cylindrical surfaces extracted from an engineering facility.

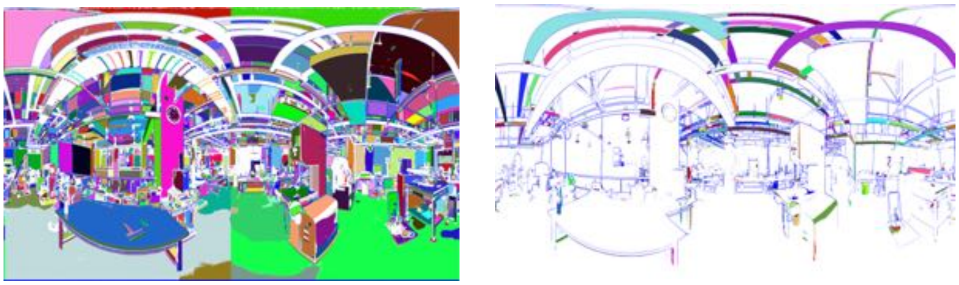

Fig. 4. Extracted surfaces. (a) planar regions, and (b) cylindrical regions.

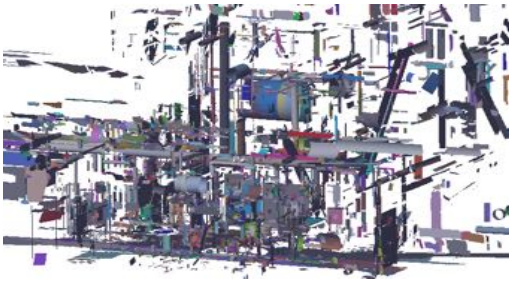

Fig.5 Planes and cylinders extracted from a point-cloud.

3.4 Reconstruction of pipe structure from cylindrical regions

In our previous work, we reconstructed pipe structures by combining cylindrical surfaces [6, 7]. In this method, coaxial cylinders are connected to reconstruct straight pipe shapes. Then we consistently connect pipes by creating connecting components, such as flanges, reducers and elbows.

While 3D models of pipes can be reconstructed only by extending cylindrical surfaces, the reconstruction of objects from planar faces is more complicated, because the boundary edges of each face has to be estimated.

4. Reconstruction of Box Shapes and Polygonal Prisms

4.1 Reconstruction of box shapes

We reconstruct box shapes by combining planar regions. In this paper, a box shape is defined as a cuboid. As shown in Fig. 6, we can consider three cases for the combinations of planar regions to reconstruct box shapes. Fig. 6 (c) shows a special case in which a face of a box is nearly perpendicular to laser pulses. In this paper, we reconstruct box shapes only when two or three perpendicular faces are extracted, because the depth of the box in Fig. 6 (c) cannot be determined. To process cases such as Fig. 6 (c), point-clouds that are captured at different positions are required.

Perpendicular faces may be separated because of chamfers, as shown in Fig. 6 (a). A single face may be partly occluded, or be divided into multiple regions, as shown in Fig. 6 (b). To robustly extract box shapes, we first roughly estimate possible combinations of planes and then estimate the consistency of the box shape.

In our method, neighbor planes are detected on the 2D map. We select a planar region, and search for neighbor planes. We enlarge the selected regions by r_i pixels from the boundary point $\boldsymbol{p_i}$. We denote the angle interval as $\Delta\phi$, and the normal of the plane as $\boldsymbol{n}$. L is a constant threshold value, which indicates the maximum length from the boundary. We specified the value L as 6 cm. Then the search range r_i can be determined as:

$$r_i = round\left(\frac{\boldsymbol{p_i}\cdot\boldsymbol{n}}{|\,\boldsymbol{p_i}\,|^2\,\Delta\varphi}L\right) \tag{3}$$

We expand the region within r_i pixels from each boundary points. When coplanar regions are detected in the expanded pixels, they are merged with the seed region. When perpendicular faces are detected, they are regarded as pair regions.

When perpendicular regions are detected, the intersection lines are calculated, as shown in Fig. 7. When three planar regions are obtained, three intersection lines are detected. The sizes of rectangles are determined so that the rectangles cover planar regions. When only two planar regions are obtained, a single intersection line is calculated. Then the two rectangles are determined so that regions are covered, and finally a box shape is obtained, as shown in Fig. 7 (b).

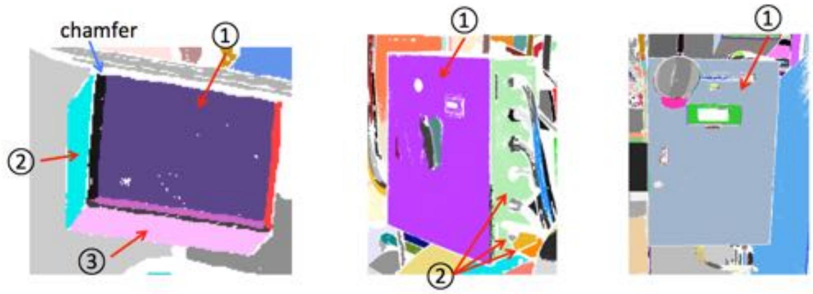

Fig. 6. The number of extracted planes. (a) three planes, (b) two planes, and (c) a single plane.

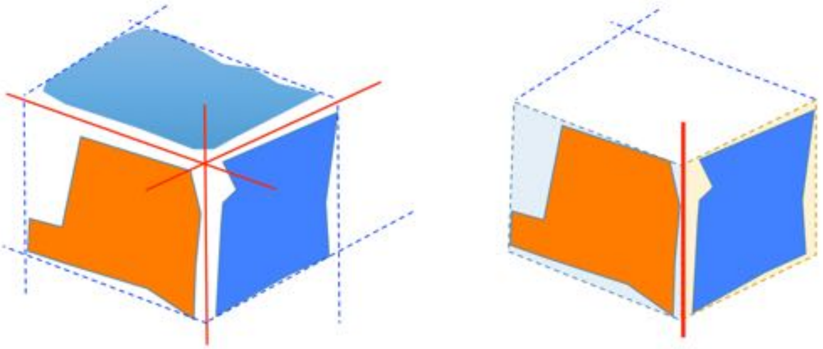

Fig. 7. Calculation of boundary edges. (a) calculation from three faces, and (b) calculation from two faces.

4.2 Visibility check of estimated shapes

In our method, when perpendicular planar regions are closely located on the 2D map, box shapes are estimated. However, false boxes may be generated from inadequate pairs of planar regions. To avoid false boxes, we apply visibility check to investigate the consistency with a point-cloud.

As shown in Fig. 8 (a), if a rectangle face is partly occluded, other objects exist between the occluded face and the laser scanner. Then measured points exist in front of the plane. If the measured point is located in the rear of the plane, the rectangle is inconsistent.

The visibility check is applied to all points $\{\boldsymbol{p}_i\}$ inside the rectangle region. As shown in Fig. 8(b), when the distance of $\boldsymbol{p}_i$ is larger than the distance of the intersection point on the plane, the rectangle face is regarded as an inconsistent face. When one or more rectangles are inconsistent, the box shape is rejected.

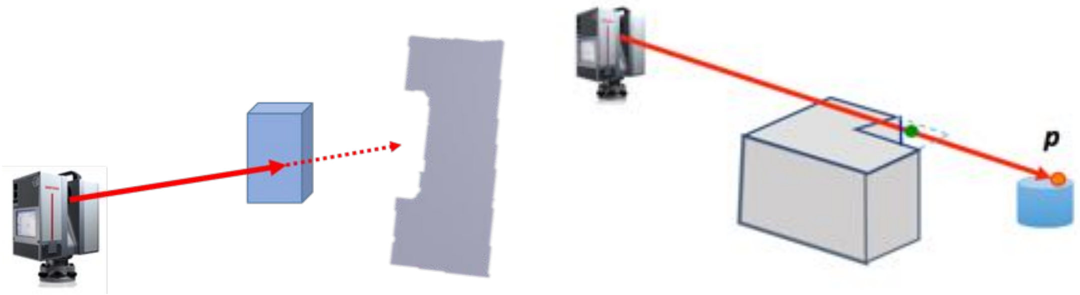

Fig. 8. Visibility check. (a) occluded by the obstacle, and (b) an inconsistent rectangle.

4.3 Reconstruction of polygonal prisms

When a box shape is rejected according to visibility check, polygons are searched on planar regions. Fig. 9(a) shows a rejected box shape. Fig. 9(b) shows a plane of the box. This face is inconsistent as a rectangle face. Then the boundary points of the planar region are extracted and straight lines are searched using the RANSAC method. In Fig. 9(b), detected straight lines are shown in blue. A polygonal face is reconstructed by calculating intersecting points between detected straight lines and the boundary edges of the rectangle, as shown in Fig. 9.

When a polygonal face is generated, a polygonal prism is created by sweeping the polygon face. When the polygonal prism does not satisfy the visibility check, it is rejected and other rectangles are tested whether a consistent polygonal prism can be created.

Fig. 10 shows an example of shape reconstruction. This point-cloud was captured from a machine room in our university. Box shapes and polygonal prisms are shown in yellow and red colors. Inconsistent shapes are shown in green.

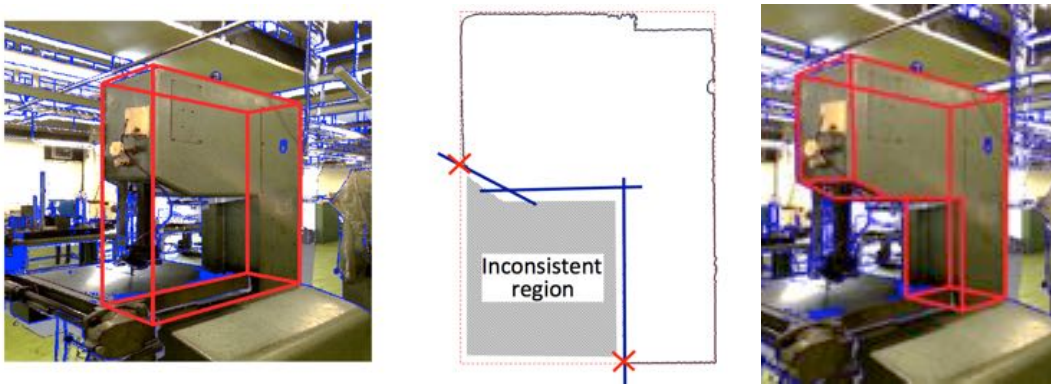

Fig. 9. Generation of polygonal prisms. (a) an inconsistent box shape, (b) generation of a polygon, and (c) a created polygonal prism.

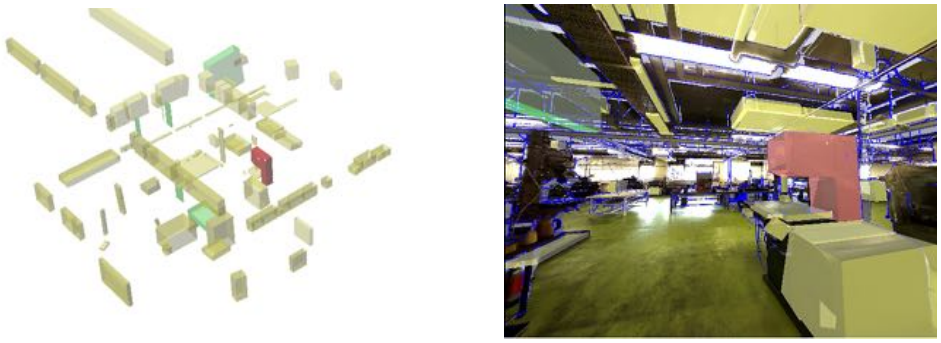

Fig. 10. Detection of box shapes and polygonal prisms.

5. Experimental Results

In the first experiment, we extracted box shapes from a point-cloud, in which we placed 14 boxes on the floor, as shown in Fig. 11(a). A point-cloud is captured points using HDS7000, which is a phase-based laser scanner developed by Leica Geosystems.

Fig. 11 (b) shows detected boxes. 13 boxes were extracted using our method. One box shown in a red circle could not be detected, because the system could extract only a single planar face from this box. Box shapes were also extracted from a sofa, but one fabric cushion was not detected as a box, because the side face was not recognized as a plane.

Table 1 shows the success rates about 14 boxes. Our method could successfully extract most box shapes. In our experiments, we used a PC with 3.4 G Hz Intel Core i7-2600 CPU with 16GB RAM.

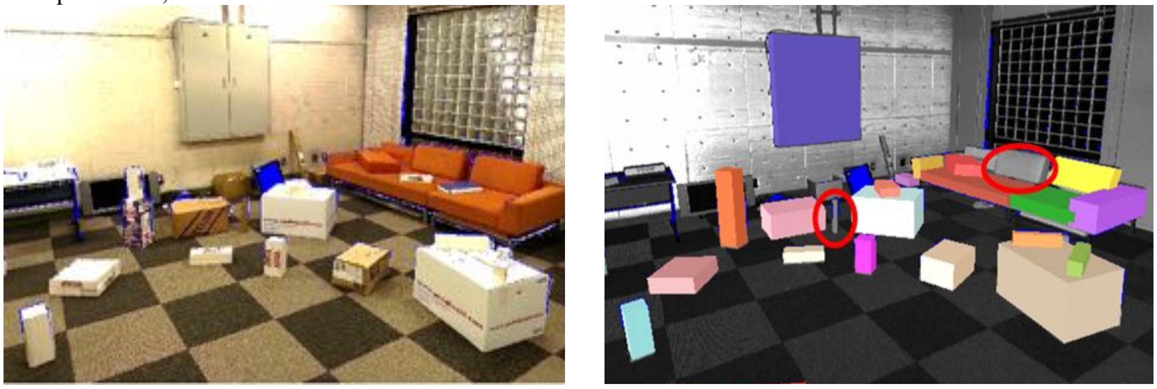

Fig. 11. Rconstruction of box shpaes. (a) points with colors, and (b) extracted box shapes.

Table 1. Detected shapes from a room at which boxes are placed.

Number of Points	10,602,562
Detected planes	1,070
Detected boxes / total number	13 / 14
Precision	100 %
Recall	92.9 %
CPU Time	10.6 sec

In the next experiment, we extracted boxes and polygonal prisms from point-clouds captured from a boiler room. The point-cloud is shown in Fig. 12 (a), and a part of detected shapes are shown in Fig. 12 (b). Table 2 shows the result. The precision and recall were more than 80%.

Our experiments show that our method can extract box shapes and polygonal prisms in a practical time.

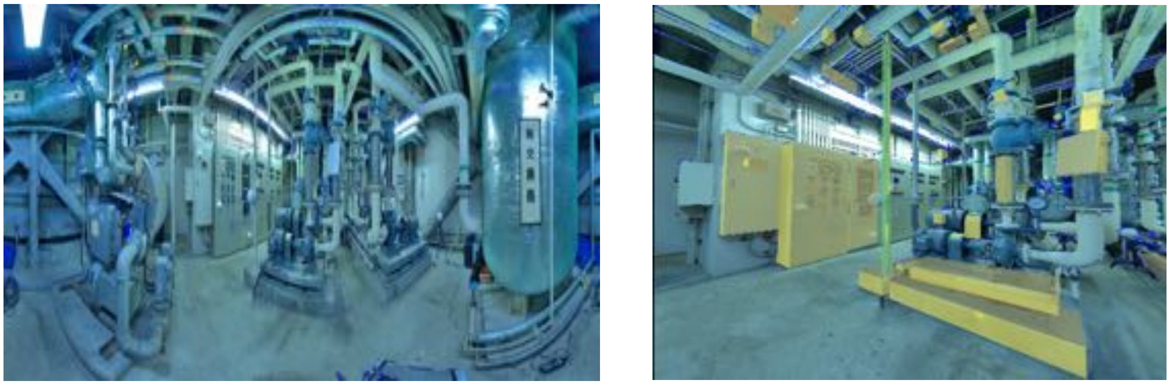

Fig. 12. Reconstructed box shape from a boiler room. (a) point-cloud, and (b) detected shapes.

Table 2. Detected shapes from a boiler room.

Point size	40,660,717
Detected planes	5,580
Detected box shapes	91
Precision	80.2% (73/91)
Recall	84.0% (68/81)
CPU Time	82.5 (sec)

6. Conclusion

In this paper, we proposed a method for reconstructing box shapes and polygonal prisms from large-scale point-clouds. In our method, planar regions are detected and they are combined so that box shapes or polygonal prisms are constructed. Detected shapes are verified using the visibility check. In our experiments, the precision and recall were about 80%.

In future work, we have to improve success rates. In addition, it is required to extend feature types that can be extracted. We would like to consider more general reconstruction method.

Reference

[1] K. Kawashima, S. Kanai, and H. Date, As-Built Modeling of Piping System from Terrestrial Laser Scanned Point Clouds Using Normal-Based Region-Growing, , in *Proc. Asian Conference on Design and Digital Engineering* (2013)

[2] J. Lee, C. Kim, H. Son, and C. Kim, Skeleton-Based 3D Reconstruction of As-Built Pipelines from Laser-Scanned Data, in *Proc. ASCE International Conference on Computing in Civil Engineering* (2012), pp. 245

[3] H. Masuda, T. Niwa, I. Tanaka, and R. Matsuoka, Reconstruction of Polygonal Faces from Large Scale Point Clouds of Engineering Plants, in *Proc. of CAD'14* (2014), pp. 150-152

[4] H. Masuda, and I. Tanaka, Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds, *Computer-Aided Design and Applications* 6 (2009), pp. 387-398

[5] H. Masuda, and I. Tanaka, As-Built 3D Modeling of Large Facilities Based on Interactive Feature Editing Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds, *Computer-Aided Design and Applications* 7 (2010), pp. 349-360

[6] R. Matsuoka, and H. Masuda, Reconstruction of Structure Shapes of Facilities from Large-scale Point Cloud: Estimation and Evaluation of Connecting Components Based on Extracted Surfaces, *Journal of The Japan Society of Precision Engineering* 80 (2014), pp. 939-943

[7] R. Matsuoka, and H. Masuda, Reconstruction of Structure Shapes of Facilities from Large-scale Point Cloud: Merging of Multiple Models and Evaluation of Reliability, *Journal of The Japan Society of Precision Engineering* 80 (2014), pp. 604~608.

[8] T. Mizoguchi, T. Kuma, Y. Kobayashi, and K. Shirai, Manhattan-World Assumption for As-Built Modeling Industrial Plant. *Key Engineering Materials* 523 (2012), pp.350-355

[9] R. Schnabel, R. Wahl, and R, Klein. Efficient RANSAC for point ‐ cloud shape detection, *Computer graphics forum* 26 (2007), pp. 214-226

[10] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, Automatic Reconstruction of As-Built Building Information Models from Laser-Scanned Point Clouds, *Automation In Construction* 19 (2010), pp. 829-843