Automatic 3D Feature Extraction from Structuralized LIDAR Data

Miao Wang
Department of Geomatics, National Cheng Kung University
No.1 University Road, Tainan 701, Taiwan
monsterr@seed.net.tw

Yi-Hsing Tseng
Department of Geomatics, National Cheng Kung University
No.1 University Road, Tainan 701, Taiwan
tseng@mail.ncku.edu.tw

Abstract: LIDAR, or laser scanning, is capable of collecting accurate 3D coordinates of scanned points densely and sub-randomly distributed on scanned object surfaces. The huge amount of 3D points implies abundant recessive spatial information which can be turned into dominant information through various data processing methods. To explore valuable spatial information from LIDAR data automatically becomes an active research topic, for example extracting digital elevation model, buildings, and trees from LIDAR data. It has long been recognized that extracting features from implicit data is the first and essential step of deriving explicit information from data. In contrast to 2D features can be extracted from image data, this paper focuses on extracting 3D features from a point cloud data set. Because the most prominent features in point cloud are co-plane points, the proposed method begins with extracting 3D plane features. Then, 3D edges and corners can be extracted by intersecting neighboring planes. Most significant 3D features can be extracted automatically through the proposed data processing method. In order to handle the large amount of sub-randomly distributed point cloud data efficiently, organizing the data set is required during the data processing. This paper proposes an octree-based split-merge-intersect method to organize LIDAR point cloud and extract 3D features. The proposed method was applied on both airborne and ground LIDAR data. The test results show the promising capability of extracting 3D features from point cloud data. The need of economic computation time demonstrates the efficiency of the developed method.
Keywords: LIDAR, Octree, Structuralize, Feature Extraction

1. Introduction

Light detection and ranging (LIDAR), or called laser scanning, is a novel technique for highly automated collection of a large amount of accurate 3D point data densely distributed on the scanned object surface [1-3]. The collected data are commonly called point cloud, range image or 3D image. A point cloud data set can be collected on the air by using an airborne LIDAR system or on the ground by using a ground-based 3D laser scanner. The huge amount of 3D points implies abundant recessive spatial information which can be turned into dominant information through various data processing and analyzing methods. To explore valuable spatial information from point cloud data automatically becomes an active research topic, for example extracting digital elevation model, buildings, and trees from LIDAR data [4-6].

It has long been recognized that extracting features from implicit data is the first and essential step of deriving explicit information from data. In contrast to 2D features can be extracted from image data, this paper focuses on extracting 3D features from a point cloud data set .It has also been recognized that surfaces play an important role in the quest of reconstructing scenes from sensory data [7, 8]. Plane extraction, therefore, is the most fundamental processing step of perceptual organization on a LIDAR data set.

Because the most prominent features in point cloud are co-plane points, the proposed method begins with extracting 3D plane features. Plane extraction from LIDAR data can be referred to the study field of range image segmentation [9, 10]. In order to handle the large amount of sub-randomly distributed point cloud data efficiently, organizing the data set is required during the data processing. This paper proposes an octree-based split-merge-intersect method to extract 3D features automatically. Firstly, the split-merge process organizes LIDAR point cloud data into clusters of 3D adjacent points which are distributed closely on a plane. The split process starts from the whole data set as the root node of an octree. The data set space

will be divided into 8 sub-spaces, represented by 8 sub-nodes, if the point cloud inside does not fit a plane patch. Each sub-node will be split continuously until the point cloud contained in the split space of the sub-node fits a 3D plane patch or is less than 3 points. After splitting, point cloud are divided into 3D patches and organized in an octree structure. By searching through the octree structure, a neighboring patches relation can be established as a look up table. The merge process can get neighboring patches from the patch relation table. Neighboring patches will be merged into a larger plane if they are similar. Then a neighboring planes relation table is established using the patch relation table again. The split-merge process is a 3D segmentation procedure results in segmented 3D planes organized in the octree structure. After 3D plane extraction, 3D edges and corners can be found by intersecting neighboring planes. Most significant 3D features can be extracted automatically through the proposed data processing method. Table 1 summarizes the inputs and outputs at each process of the method.

Table 1. Inputs and outputs at each process

Process	Input	Output
Split	● Point cloud	● Patches with boundaries ● Octree structure ● Neighboring patch relation table
Merge	● Patches ● Neighboring patch relation table	● Merged Planes with refined boundaries and TIN ● Neighboring plane relation table
Intersect	● Planes ● Neighboring plane relation table	● Edges and corners

The proposed method was applied on both airborne and ground LIDAR data. The test results show the promising capability of extracting 3D features from point cloud data. The need of economic computation time demonstrates the efficiency of the developed method.

2. 3D Feature Extraction Based on Octree Structure

1) Split process

The split process starts from the whole data set as a root node of an octree. The data set space will be divided into 8 sub-spaces, if the data set could not pass the plane fitting test. Figure 1.a shows an example of octree structure. The split generates 8 sub-nodes (figure 1.b) representing the split spaces. Each sub-node will be split continuously until the scan points contained in the split space of the sub-node are distributed close to a 3D best-fit plane or less than 3 points.

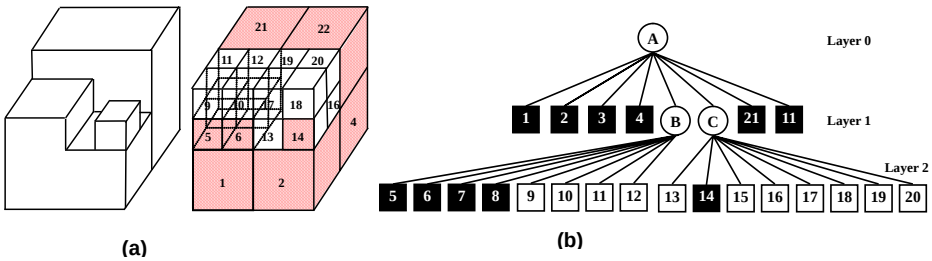


Fig. 1. Octree structure (a) divided sub-spaces (b) the tree representation

During this process, point sequence in a node will be replaced in the order of its sub-nodes. So at the end of the split process, the point cloud data will be sorted by the octree structure. Besides, sub-nodes contain no point in it will be dropped and does not be recorded. Therefore, it can reduce the storage space of an octree structure.

1.1) **Calculating fitting plane**

The best-fit plane in each sub-node is determined using least-squares estimation, i.e., minimizing the squares sum of the distances from points to the fitting plane. In the 3D Euclid space, a 3D plane can be formulated as follows:

$$Ax + By + Cz + D = 0 \tag{1}$$

The distance (d_i) from the ith point $P_i(x_i, y_i, z_i)$ to the plane can be expressed as:

$$d_i = F(A,B,C,D) = \frac{|Ax_i + By_i + Cz_i + D|}{\sqrt{A^2 + B^2 + C^2}} \tag{2}$$

Then, the best-fit condition of minimizing the squares sum of the distances will be:

$$\sum_{i=1}^{N} d_i^2 \Rightarrow \min \tag{3}$$

In fact, a 3D plane has only three independent parameters. The 4 parameters (A, B, C, D) in Eq.(1) are dependent. It means for the same plane of Eq.(1) can also be formulated as follows:

$$k(Ax + By + Cz + D) = 0, k \neq 0 \tag{4}$$

The 4 parameters [kA, kB, kC, kD] represent the same best-fit plane. So the iterative process of least-squares estimation will not converge. An extra constraint equation is needed to solve one unique solution. The constraint can be introduced by setting the normal vector of the 3D plane be a unit vector. It can be formulated as:

$$G(A,B,C,D) = A^2 + B^2 + C^2 = 1 \tag{5}$$

Under the circumstance, Eq.(2) can be revised as：

$$d_i = |Ax_i + By_i + Cz_i + D| \tag{6}$$

To solve the combined equations of (3) and (5) is a standard least-squares problem with constraints. However, the objective functions are nonlinear. When the Newton's method is applied to solve the least-squares problem, it needs initial approximation values of the unknown parameters (A, B, C, D). The follows descript how to determine the initial approximation values.

For a plane $Ax + By + Cz + D = 0$, the 3 plane parameters (A, B, C) represent its normal vector and will always at least has one dose not equal to 0. For example, if A ≠ 0, it will always exist a none-zero constant ***k*** to let ***k***A=1. The plane equation can be formulated as follows:

$$x + by + cz + D = 0 \tag{7}$$

where

$$b = kB\ ,\ c = kC\ ,\ d = kD$$

In order to determine the plane parameter, Eq.(7) can be formulated as follows:

$$ax + by + cz + d = 0$$

Setting constraint as:

$$a = 1$$

The least-squares estimate model will be list as follows:

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ v_c \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & & & & \\ & 1 & & 0 & \\ & & \ddots & & \\ & 0 & & 1 & \\ & & & & \infty \end{bmatrix}, \quad V = AX + L \tag{8}$$

Eq.(8) is linear, so that the least-squares solution to the parameters can be obtained without the need of the iterations.

It has to be known which parameter of normal vector (a, b, c) of the plane is indeed dose not equal to 0 to set as the constraint when using the above adjustment method. In order to find the constraint, 6 points which has the maximum and minimum coordinate in X, Y and Z axis are selected and are denoted as $Px\max$, $Px\min$, $Py\max$, $Py\min$, $Pz\max$ and $Pz\min$. Compare the 3 distances, $\overline{P_{x\max}P_{x\min}}$, $\overline{P_{y\max}P_{y\min}}$ and $\overline{P_{y\max}P_{y\min}}$ and select the 2 points of the longest distance. Then search the third point which has the longest distance to the previous selected 2 points from point cloud. These 3 points can form a 3D plane and its plane parameters can be determined easy. In fact, these plane parameters can be use as the approximate values of the least-squares estimation, but using Eq.(8) can get a better ones. To decide which parameter to be the constraint, compare the absolute value of (A, B, C) and choose the largest ones. Figure 2 shows the idea.

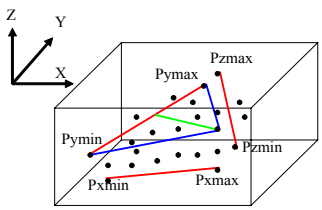


Fig. 2. The idea of selecting constraint parameter.

The distance residuals can be calculated after the plane parameters are solved. Split process will proceed continuously if there is a residual larger than the preset threshold, otherwise a plane is formed.

1.2) Density of the point cloud on a fitting plane

The point cloud in a sub-space may not distribute evenly on the fitting plane. If the fitting plane is assumed to represent a scanned ground or object surface, the point cloud should distribute evenly on the plane. In order to find planes corresponding to the distribution of point cloud, distribution density is checked for further splitting. For example, unbalanced distribution of points in figure 3 will show a low distribution density. Split can proceed further, until the distribution density of point cloud on each fitting plane is higher than a preset threshold.

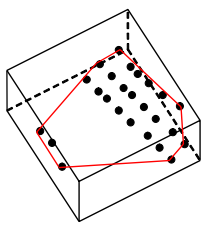
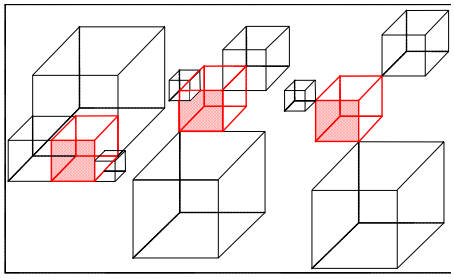


Fig. 3. An example of unbalanced distribution of points

To determine the distribution density, the area of the plane should be calculated first. In order to calculate the area of the plane, the segmented points are used to construct a TIN (Triangular Irregular Network) structure. The points are projected onto the plane, and then rotate to horizontal to obtain two dimensional distributed points, which can be used to construct Delaunay triangles. The outer border of the TIN then will be the boundaries of the convex polygon containing the segmented points. So the area of the plane can be calculated.

1.3) Establish neighboring patches relation table

The best-fit plane in a node generated in the split process will be called as a "patch" in the following to distinguish with the larger plane generated in the merge process. This process split LIDAR point clouds into patches base on octree. In the octree structure one node has 3 types of neighboring node which can be express as share face, share edge and share vertex [11]. Fig. 4 shows the 3 types of neighboring nodes of greater, equal and smaller size of a node. The octree structure allows us to find all the 3 types of neighboring nodes (patches) easily to establish a neighboring patch relation look up table by using node-searching algorithms. The neighboring patch relations look up table than can be the input data of the merge process.



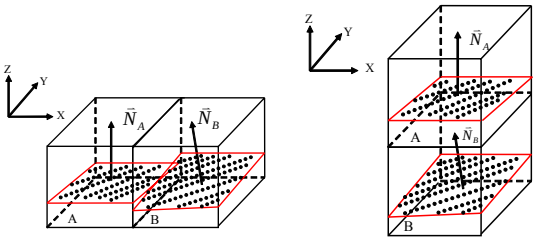
(a) (b) (c)

Fig. 4. 3 types of neighboring node (a) share face (b) share edge (c) share vertex

2) Merge process

A merge process is needed to connect neighboring 3D patches which are similar. Firstly, all leaf nodes (contain no sub-node) in the octree are sorted by their area of the contained plane. The merge process starts from the node contained the largest patch. The neighboring patches are retrieved from the neighboring patch relation look up table and the merge process will first check whether their normal vectors are similar. Fig 5.a shows the idea of checking normal vectors. However, one may find neighboring planes show in Fig. 5.b having similar normal vectors. To solve this situation, additional check of the plane relation is needed. After two neighboring planes are found similar, the rigorous calculation will be triggered again to recalculate the parameters of the merged plane and to ensure the merge availability. The merge process can be done iteratively by searching and checking neighboring planes to generate a larger plane.

This process merges patches into 3D plane clusters. A neighboring plane relations look up table is established from the relation of planes and their containing patches and neighboring patch relation table.



(a) (b)

Fig. 5. Similarity of two neighboring nodes A and B (a) available for merge (b) not available for merge

3) Intersection Process

In the merge process, point clouds data are segmented into 3D plane clusters. Neighboring planes can be found from neighboring plane relation look up table to perform intersection process to generate edges and corners. Edges and corners can be generated by intersect neighboring 2 planes or 3 planes respectively. Fig. 9 shows the concept of generating edges and corners form neighboring planes.

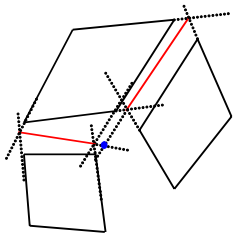


Fig. 9. Concept of generating edges and corners form neighboring planes.

3. Description of Extracted Features

The split-merge-intersect procedure extract 3D features from sub-distributed 3D point cloud data. The split-merge segmentation process results in the parameters of extracted 3D planes and segmented point clusters organized in the octree structure. The plane parameters comprise the basic descriptions of an extracted plane, such as orientation or gradient. Further descriptions can be derived from the distribution of segmented points, for example the plane boundary, area, center point, point density, surface roughness, and average intensity. Further analysis or classification of extracted planes followed based on those descriptions.

Fig. 6 shows a set of segmented points, and Fig. 7 shows the constructed TIN structure. In order to find a boundary which closely encompasses the point cluster, the TIN structure can be refined by eliminating large triangles. Fig. 8 shows the refined boundary.

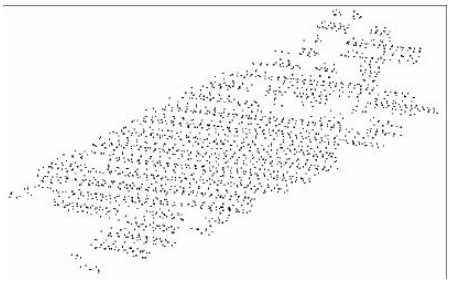


Fig. 6. A set of segmented points

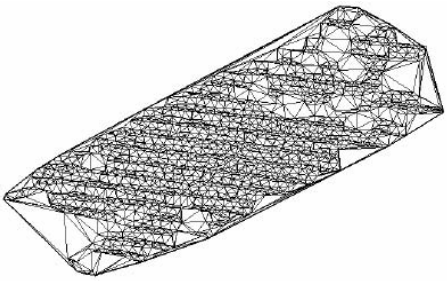


Fig. 7. The constructed TIN structure.

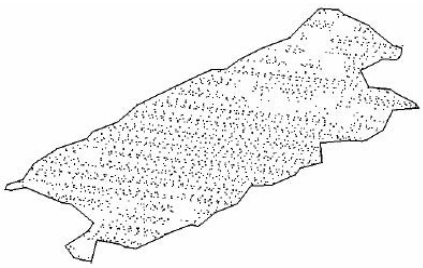


Fig. 8. The refined boundary of the extracted plane.

Edges and corners extracted in the intersect process can be regard as the frame line and vertex of the

scanned object. Because the edges and corners are generate by intersecting the best-fit plane of the object, they are meaningful in 3D model reconstruction application. Fig. 9. shows 3 edges and 1 corner extracted form a gable roof.

Fig. 9. Edges and corner extracted from a gable roof.

4. Experiments

The proposed method is applied to both airborne and ground LIDAR data. The test data include an airborne LIDAR data set collected in Hsinchu, Taiwan with Leica ALS40 and a ground LIDAR data set obtained in the Library of National Cheng Kung University, Tainan, Taiwan with Optech ILRIS-3D laser scanner.

The airborne test data is a building with gable roof. The point number is 15,026 pts. and the point density is about 1.7 pts./m^2. It takes 0.573, 2.715 and 0.054 second in the split, merge and intersect process. Fig. 10 shows the point distribution, in which the rainbow colors from violet to red represent the differences of point height from low to high. After splitting process, the points are segmented into clusters of octree subspaces, in which the contained points are distributed nearby a 3D plane. Fig. 11 shows the segmented points and the plane boundaries resulted from the split process. Neighboring planes are merged, if they are similar. After merging process, a TIN structure can be formed for each cluster of segmented points and the refined boundary of the extracted plane can be found. Fig. 12 shows the TIN structures and the boundaries of extracted planes whose areas are larger than 50 m^2. Fig. 13. shows the edges and corners extracted by intersecting planes whose areas are larger than 50 m^2.

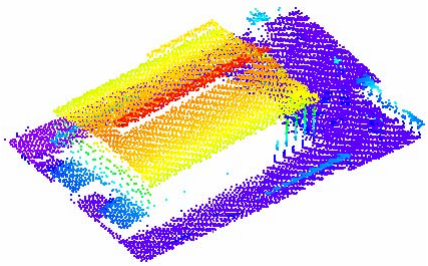


Fig. 10. The test point cloud data.

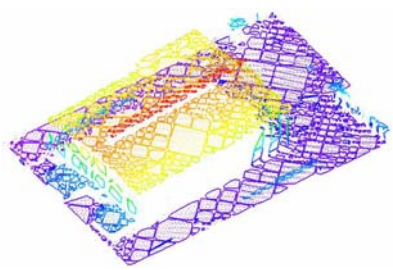


Fig. 11. Segmentation of points after the split process.

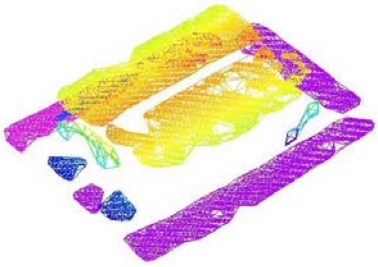


Fig. 12. TIN Structure and boundaries of extracted plane

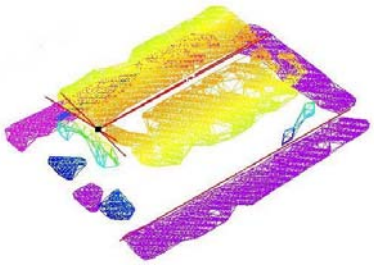


Fig. 13. Extracted edges and corners.

The ground test data is a part of data cut from the point cloud of the library of National Cheng Kung University. The point number is 105,070 pts. and the point density is about 714 pts./m2. It takes 2.651, 55.442 and 0.128 second in the split, merge and intersect process. Fig. 14 shows the point distribution. Fig. 15 shows the segmented patches and the plane boundaries resulted from the split process. The boundaries of extracted planes whose areas are larger than $5m^2$ are depicted in Fig. 16. The extracted edges and corners in the circle area are showed in Fig. 17.

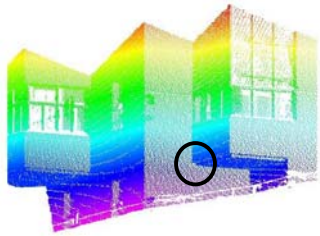


Fig. 14. The test point cloud data.

Fig. 15. Segmentation of points after the split process.

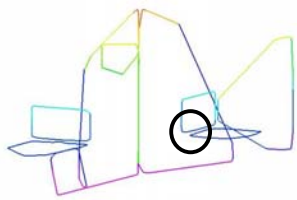


Fig. 16. Boundaries of extracted planes

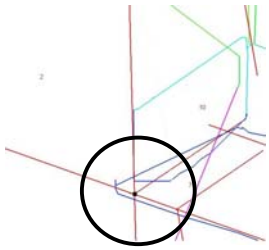


Fig. 17. Extracted edges and corners.

5. Concluding Remarks

LIDAR data indeed are 3D coordinates of a set of sub-randomly distributed points. Traditional image processing techniques do not fit the needs of handling LIDAR data. The octree structure is an appropriate data structure to organize 3D distributed LIDAR points. This paper proposes a 3D split-merge-intersect method base on octree structure to extract 3D features from LIDAR point cloud automatically. The LIDAR points can be clustered into point groups which form 3D planes and are organized in an octree structure. The split-merge process is equivalent to a 3D segmentation of the LIDAR data to extract plane surfaces. The intersect process generates 3D edges and corners by intersecting neighboring 2 and 3 planes. This method can be applied to both airborne and ground LIDAR data. The examples show the need of economic computation time and demonstrate the efficiency of the developed method.

We also expect that this process and the extracted 3D features would benefit many applications of LIDAR data. For example:

- Data filtering and compression – points were not used to form planes can be filtered out and densely distributed on a plane can be reduced.
- Data merge and adjustment – the extracted 3D features include planes, edges and corners can be used as control surfaces or conjugate features to connect data sets collected from different scans (strips).
- Texture mapping – digital images or textures of the scanned surface can map onto the extracted planes to form a photo-realistic view.
- 3D model reconstruction – the octree structure can provide the topology of the extracted 3D features to reconstruct 3D model.

Acknowledgement

This research is sponsored under the grants of the project “NSC 93-2211-E-006-066”. The authors deeply appreciate for the support from the National Science Council of Taiwan and sincerely express our gratitude to the Council of Agriculture for the provision of airborne LIDAR data.

REFERENCES

[1] F. Ackermann, 1999. Airborne Laser Scanning - Present Status and Future Expectations, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54: 64-67.

[2] P. Axelsson, 1999. Processing of Laser Scanner Data -Algorithms and Applications, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54: 138-147.

[3] A. Wehr and U. Lohr, 1999. Airborne Laser Scanning -an Introduction and Overview, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54: 68-82.

[4] N. Haala and C. Brenner, 1999. Extraction of Building and Trees in Urban Environments, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54: 130-137.

[5] G. Vosselman and S. Dijkman, 2001. 3D Building Model Reconstruction from Point Clouds and Ground Plans, *International Archives of Photogrammetry and Remote Sensing*, Annapolis, Maryland.

[6] J. Shan and A. Sampath, 2005. Urban BEM Generation from Raw Lidar Data: A Labeling Algorithm and Its Performance, *Photogrammetric Engineering and Remote Sensing*, 71: 217~226.

[7] X. Jiang, H. Bunke, and U. Meier, 2000. High-level feature based range image segmentation, *Image and Vision Computing*, 18: 817-822.

[8] I. Lee and T. Schenk, 2001. 3D Perceptual Organization of Laser Altimetry Points, *International Archives of Photogrammetry and Remote Sensing*, Annapolis, MD.

[9] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, 1996. An Experimental Comparison of Range Image Segmentation Algorithms, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18: 673-689.

[10] K. Koester and M. Spann, 2000. MIR: An Approach to Robust Clustering–Application to Range Image Segmentation, *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 22: 430-444,

[11] Samet, H., 1990.*Application of Spatial Data Structures: Computer Graphics, Image Processing, and GIS,* Addison-Wesley Publishing Company, pp.57-110.