

# A Mixed Algorithm for Building Facade Extraction from Point Clouds

Rui Zhang<sup>a,b,\*</sup>, Jiayi Wu<sup>b</sup>, Guangyun Li<sup>a</sup>, and Li Wang<sup>a</sup>

<sup>a</sup>North China University of Water Resources and Electric Power, Zhengzhou, 450045, China

<sup>b</sup>Information Engineering University, Zhengzhou, 450001, China

---

## Abstract

As a leading method for capturing 3D urban scene data, laser scanning technology has been increasingly used in feature extraction, object recognition and modeling tasks. This study presents a new strategy that can be used to quickly and accurately extract building facade features based on point clouds, which are captured from laser scanners. The data first need to be pre-processed, including building a Kd-OcTree mixed index and calculating the normal vectors of point clouds using principal component analysis (PCA). On this basis, the initial clusters are obtained via fuzzy clustering, and then the generalized Hough transformation (GHT) is used in each cluster according to the sampling interval to detect the local peak value to obtain the preliminary plane. Next, similar planes are merged together based on the normal vectors and distance thresholds of the pending planes to generate better planeness. Finally, the extraction effects are optimized by the adjunctive judgment of neighborhood points, which is used to classify boundary points into the correct plane. The proposed approach has been tested with three different terrestrial laser scanners (TLS) datasets, and the results show that this mixed approach is able to speed up building facade extraction as well as the recall.

**Keywords:** building facade extraction; fuzzy clustering; 3D generalized Hough transformation; normal vector of point clouds; similar patches merging; re-judgment of coplanar points

(Submitted on March 7, 2018; Revised on April 19, 2018; Accepted on May 25, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

Laser scanning technology using non-contact measurement methods is able to obtain rich three-dimensional surface data directly, including 3D geographical coordinate information and spectral data of scanned ground objects. The process of data collection conducted by laser scanning technology is not affected by the environment, and it provides advantages such as high scanning speed, abundant data, and high precision and satisfactory real-time performance. It has been widely applied in fields such as city planning, GIS map updating, intelligent transportation, historic preservation, and virtual reality to detect and segment real scenes rapidly and accurately based on 3D laser scanning data, and it has become a hot and difficult topic. As the most basic and vital composition in a 3D city scene, building feature extraction based on laser scanning point clouds has been a major concern in fields such as photogrammetry and remote sensing, computer vision, and image processing.

In recent years, there has been a lot of research on building facade feature extraction. Relative algorithms can be divided into four categories. (1) Model-matching algorithms. The literature [5,20] has established a rich building facade semantic knowledge database according to the spatial topological relations and the geometric attributes of buildings. Then, the segmented fields are progressively judged according to the semantic rules, and the right construction will be the one that meets all requirements. This algorithm requires a large amount of prior knowledge when establishing the model knowledge database; it is a semi-automatic method that requires manual judgment and yields low data processing efficiency. (2) Dimension reduction algorithms. The literature [17,18,21,22] projects the entire 3D point cloud to a two-dimensional space and uses a 2D image processing method to extract features to reduce the amount of data and lower the difficulty of data processing. Meanwhile, massive space information is lost, which causes more difficulties in identifying each building facade automatically, rapidly, and accurately. Furthermore, if point clouds are occluded seriously, the effect of facade extraction will be even worse. (3) Clustering segmentation algorithms. Previously published methods are based on a

---

\* Corresponding author.

E-mail address: [zhangrui@ncwu.edu.cn](mailto:zhangrui@ncwu.edu.cn)

clustering algorithm [2,3,4,13,14], which are simple and highly automated, but they can only implement rough classification of the building facades. The boundary points and coplanar points cannot be accurately classified to the corresponding facade. (4) Statistical analysis algorithms. The Random sampling Consensus algorithm (RANSAC) [7,11] is a type of robust planar matching method. However, it can be severely affected by factors such as the select rule of random points and the distance threshold of point to plane. As result, it is likely to generate over-segmentation of the parameter plane and misjudgment of the location of the coplanar points, as well as fake planes. The region growing algorithm [19] is greatly influenced by the location of the initial seed point, and it is difficult to achieve rapid extraction of building facades; usually used for preliminary semantic segmentation, it is necessary to combine it with other methods during the fine processing phase. A previously published algorithm [1,8,12,23] needs to transform all point clouds from image space to parameter space, which requires increased time. Moreover, the processing speed drops exponentially with the increase of point cloud data. Therefore, the automatic rapid extraction of building facade information based on 3D laser point clouds still faces great challenges.

## 2. Algorithm description

Figure 1 is a rapid extraction fusion algorithm flowchart for building facade features based on 3D point clouds. First, the Kd-ocTree mixed index is established to organize the 3D point clouds to improve the speed of building facade extraction. Each point cloud adopts PCA to calculate normal vectors and complete data pre-processing. Then, fuzzy clustering is conducted according to the topological relationship and spatial relationship of points on the same facade of the building to obtain multiple facade clusters and preliminarily extract building facades. For each facade point cluster, the GHT is conducted according to the normal vector, and peak detection is carried out to detect the features of the building facade. This method effectively avoids multiple peak points, misjudgment, or other phenomena that often occur when directly conducting peak detection in Hough transformation accumulator space by 3D coordinates. Finally, the plane extraction effect is optimized, including similar patch merge and re-judgment of coplanar points.

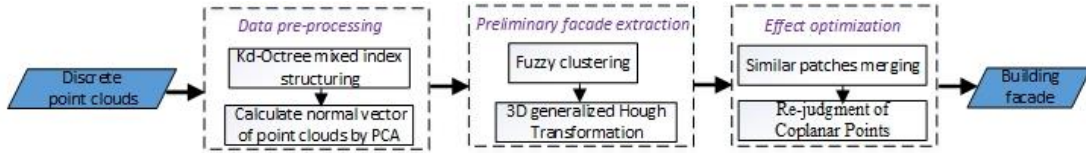


Figure 1. Mixed algorithm flowchart on rapid extraction of building facade features

### 2.1. Data pre-processing

Data pre-processing involves two operations: structuring a mixed index and calculating the normal vectors of point clouds. First, to efficiently organize the massive point cloud data and quickly retrieve them in blocks, a highly efficient index structure for the original point clouds is needed. In this study, the index structure of the Kd-ocTree mixed index structure is designed. This index structure rebuilds the neighborhood relationship among point clouds according to the dimension of segmentation during the KD tree construction period, avoiding a lean of the tree structure, and fundamentally solves the problem of slow neighborhood search speed, ensuring rapid extraction of building facades. The details of the Kd-ocTree index can be found in our early work [24].

On the basis of the mixed index, first calculate the normal vectors of the point clouds and conduct fuzzy clustering according to the normal vectors obtained. To calculate the vector information quickly and effectively, and inspired by the point cloud vector estimation method applied in previous studies [9,25], the current study sets the distance threshold of point clouds to pre-samplings of the processing point sets and conducts a K nearest neighbors (KNN) search on pending point clouds. Then, a pending point and its neighbor points are used to calculate its initial value of the normal vector by the PCA method; according to the constraint condition, the dot product of the normal vector of each point  $\vec{n} = (n_x, n_y, n_z)$  and incidence direction of the laser  $\vec{r} = (x, y, z)$  must be less than zero  $\vec{n} \cdot \vec{r} < 0$  to adjust the orientation of the normal vector, and then the final normal vector is obtained.

### 2.2. Preliminary plane extraction

#### (1) Fuzzy clustering

To segment each facade into independent semantic information rapidly and effectively before accurately extracting building facades, it is necessary to use a combination of normal vector angle and Euclidean distance clustering to roughly segment building facades. For segmentation of boundary points, this method can be more accurate than the traditional Euclidean

distance method of clustering because, in addition to considering the distance among points, the normal vector is also considered, and therefore it is able to divide those points, which are nearby in distance but belong to two different planes, into different facades.

This method clusters point clouds in each local Octree node according to the Kd-Octree index structure, which divides the total point clouds into many data blocks equally according to the segmentation strategy, to speed up data processing. According to a certain constraint condition, points in the same cluster can be classified into the correct patches preliminarily, which effectively avoids the phenomenon of insufficient segmentation; however, there will be an over-segmentation phenomenon, which results in a lot of litter clusters. Thus, patch merging is indispensable later on. Figure 3 shows the target clusters obtained after fuzzy clustering under different constraint conditions.

In this study, there are several user-defined thresholds that have a crucial impact on the quality of results obtained. In this section, we set four thresholds: number threshold of points in global KD leaves node, number threshold of adjacent points, angle threshold, and distance threshold. The latter two thresholds are the most important parameters which are used for GHT and patch merging, as well as in this section. Thus, we analyze their impact on the performance in section 3.3. Here, we discuss how the first two thresholds are determined.

**Number threshold of points in global KD leaves node:** In the process of constructing global KD index, the number threshold of points in global KD leaves node  $KdBlockPtsThreshold$  is needed. If the point number of current node is less than  $KdBlockPtsThreshold$ , the current node is a leaf node. The concrete constructing process was introduced in our early work [24]. Taking Sample data 1 (75461 points) for example, when  $KdBlockPtsThreshold=10000$ , the global KD tree is shown as Figure 2. The point numbers of nodes in the same layer are nearly same, which ensures that the KD tree is a balanced tree as a whole and avoids the lean of the tree structure, and fundamentally solves the problem of slow neighborhood search speed. Still taking Sample data 1 for example, the consumed time of preliminary clustering based on Kd-Octree mixed index is 26.184s, while the consumed time is 34.295s based on Octree Index. The speed has increased 23.65%.

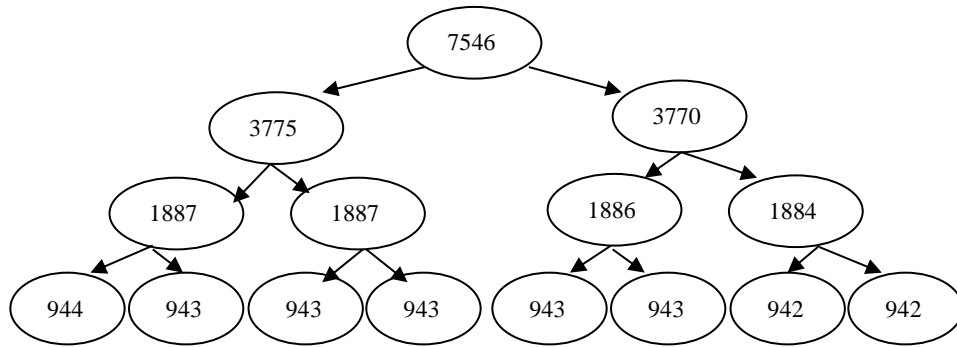


Figure 2. Example of global Kd tree

The value of  $KdBlockPtsThreshold$  is closely related to the number of points. It increases when the number of points increases. For the same point cloud, the smaller the value of  $KdBlockPtsThreshold$ , the more leaf nodes of the global Kd tree and the higher the number of clusters. Taking Sample data 1 for example, the number of clusters is 991 when  $KdBlockPtsThreshold=10000$ , and the number of clusters is 1009 when  $KdBlockPtsThreshold=8000$ . The clustering effect is shown in Figure 3.

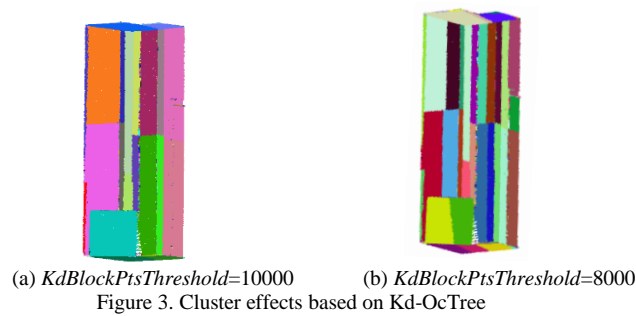


Figure 3. Cluster effects based on Kd-Octree

**Number threshold of adjacent points :** Neighborhood searching is an important ring of fuzzy clustering, where neighborhood searching radius *SearchRadius* and number threshold of adjacent points *NearPointThreshold* are requisite. Regarding *SearchRadius*, our proposed algorithm and the other four comparison algorithm all set *SearchRadius*=0.02. In this section, we only analyze the effect of *NearPointThreshold* on clustering effects. The bigger the *NearPointThreshold*, the lower the number of clusters. However, the number of neighborhood searching will be higher and the searching time will be longer. Experimental results are shown in Figure 4, where the number of clusters decreases to 429 while *NearPointThreshold* increases from 10 to 20, which reduces 56.7%, but neighborhood searching time increases from 0.575s to 1.191s. While *NearPointThreshold* increases from 20 to 30 and 40, the two changes are gentler. When *NearPointThreshold* increases to 100, the number of clusters is very small and neighborhood searching time is much longer. Overall, considering number of clusters and neighborhood searching time, this study sets *NearPointThreshold*=20.

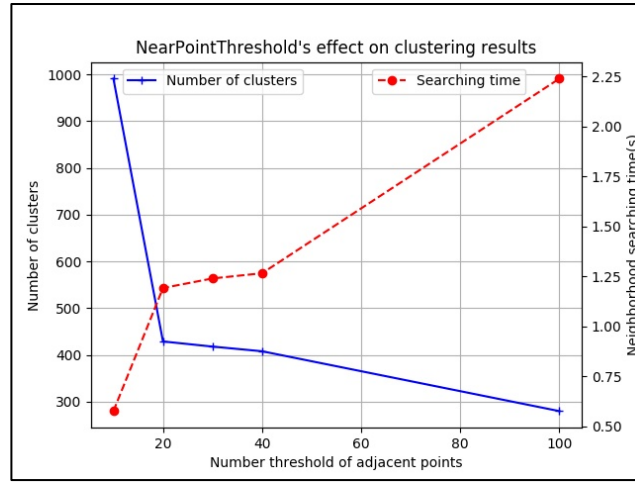


Figure 4. NearPointThreshold's effect on clustering results

## (2) FC-GHT algorithm

The Hough transform algorithm comes in three variants: standard Hough transformation (SHT), random Hough transformation (RHT), and generalized Hough transform (GHT). The SHT algorithm [6,16] yields the problem of greater computation and larger memory consumption, which is not suitable for massive point cloud data processing. RHT [26] extracts a pixel randomly among multiple pixels and maps it to the parameter space; this avoids the greater calculation and excessive memory consumption, but this also generates plentiful invalid accumulation due to random access to the sampling points and reduces the efficiency of the algorithm. GHT [15] extracts multiple pixels of a graphic according to certain rules and then maps a selected pixel to parameter space to vote; at present, it is mainly applied to feature extraction of linear, circular arcs, etc., in 2D images. The current study takes advantages of RHT and GHT and sets a sampling interval in the image space according to the number of points and their density. This approach avoids the accumulation of invalid points and improves the efficiency of the algorithm while reducing the computation and memory overhead. The current study names the algorithm, which first constructs the index structure, calculates the normal vector of the point cloud, and then extracts the building facade feature that is combined by a fuzzy cluster and GHT, as FC-GHT (Fuzzy Clustering and GHT).

For GHT, the equation of the plane characteristic that is obtained in the space polar coordinate system is:

$$\rho = x \cos \theta \cos \varphi + y \sin \theta \cos \varphi + z \sin \varphi \quad (1)$$

In Equation (1), the parameter  $\theta$  represents the angle between the projection of the normal vector  $n$  on the plane  $xOy$  and the  $x$ -axis,  $\varphi$  represents the angle between it and the projection of the normal vector  $n$  on the plane  $xOy$ , and  $\rho$  represents the distance between the original point  $O$  and the plane. The formulas for converting the unit normal vector of each point  $n = \{n_x, n_y, n_z\}$  to angles  $\theta$ ,  $\varphi$  and distance  $\rho$  are  $\theta = \arctan(n_y / n_x)$ ,  $\theta \in [0^\circ, 360^\circ]$ ;  $\varphi = \arcsin(n_z)$ ,  $\varphi \in [-90^\circ, 90^\circ]$ ; and  $\rho = d$ .

Next, divide the voting space. Set the segment width of  $\rho$ ,  $\theta$  and  $\varphi$  as *Width*, *Theta* and *Phi*, respectively, and the

range of the values of  $\rho, \theta$  and  $\varphi$  in the parameter space is divided evenly as  $D = (d_{\max} - d_{\min}) / \text{Width}$ ,  $T = 360 / \text{Theta}$  and  $P = 180 / \text{Phi}$ .  $d_{\max}$  and  $d_{\min}$  stand for the maximum value and minimum value of the distance between points to plane. Then, set up a 3D cumulative array  $\text{Vote}(\rho, \theta, \varphi)$  with size equal to  $D * T * P$ . At the same time, initialize all array elements to 0. Then, apply Equation (2) to vote  $\rho, \theta$  and  $\varphi$ , which are transformed from points in the point cluster. The threshold of the minimum vote is set to detect the local peak value. If  $\text{Vote}(\rho_i, \theta_j, \varphi_k) > \text{VoteThreshold}$ ,  $\text{Vote}(\rho_i, \theta_j, \varphi_k)++$ . Inverse mapping the peak point detected to image space yields the preliminary plane parameters  $a, b, c$  and  $d$ . Suppose there are two planes  $\text{plane1}, \text{plane2}$ ; if  $\overrightarrow{NV\_plane1} \cdot \overrightarrow{NV\_plane2} > \text{AngleThreshold}$  &  $|\text{Dist}_{\text{plane1}} - \text{Dist}_{\text{plane2}}| < \text{Width}$ , then combine the two planes to generate better planeness.

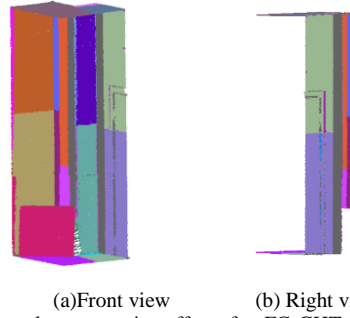


Figure 5. Preliminary plane extraction effect after FC-GHT (the number of patches is 24)

Figure 5 shows the segmentation effect after FC-GHT, and the number of patches is 24. From the plane extraction effect, it can be seen that small clusters are avoided to a certain extent. By fuzzy clustering, we get the initial clusters, and on this basis, the data processing speed can be improved in two ways: (1) set sampling rules to reduce the data size processed; (2) for each point cluster, GHT is carried out for local peak detection. Moreover, while GHT is being conducted, the intersection angle of the normal vectors and the distance between two planes need to be considered together so that the re-matched plane will have better planeness, thus ensuring the accuracy of the extracted plane.

In this section, there are three user-defined parameters: angle threshold, distance threshold, and number threshold of minimum votes. The first two use the same values as those used in fuzzy clustering. Thus, here we only discuss how to determine the threshold of minimum votes.

**Minimum vote threshold:** This threshold,  $iThreshold$ , belongs to GHT, based on which to detect the peak in parameter space. It is closely related to point numbers and has little effect on extracting results. If  $PointSize < 100$ ,  $iThreshold = 10$ ; if  $PointSize < 10000$ ,  $iThreshold = 50$ ; if  $PointSize < 1000000$ ,  $iThreshold = 100$ ; else  $iThreshold = 200$  where  $PointSize$  indicates point numbers and  $iThreshold$  indicates the minimum votes threshold.

### 2.3. Extracted plane effect optimization

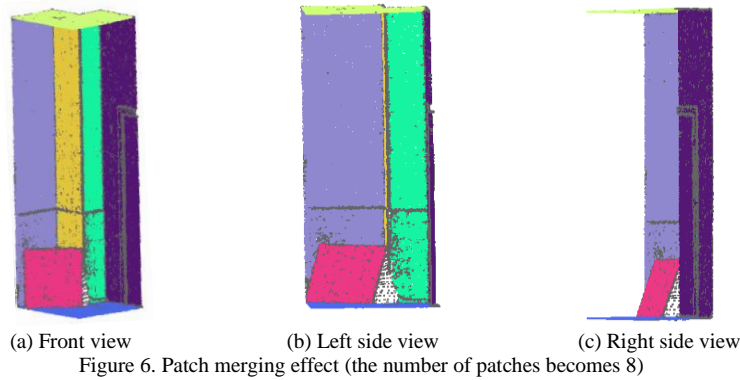
#### (1) Patch merging

The preliminary clustering and FC-GHT are both done in the global KD leaves node, influenced by the threshold of the number of point clouds of the global KD leaves node, as well as the angle threshold and distance threshold when preliminarily clustering. This ensures that the points that originally belong to the same plane are classified into different leaf nodes or in different clusters. When we extract the plane basing on each cluster, over-segmentation may occur; therefore, it is necessary to merge the similar planes and delete repeated planes. The overall approach of patch merging is summarized in Table 1.

Figure 6 shows the plane segmentation effect after patch merging; the contour is clear, the boundary between surfaces is clear, and the boundary of the door frame is clearly visible. Eight planes are extracted in total. However, for boundary points, due to the normal vectors being distributed loosely, which cannot meet the normal vector Angle threshold, those points clouds are not classified to the corresponding plane. With 75461 totals scan points, 64947 were detected, achieving a recall of 86.1%. The missing points are mainly boundary points and frame points; these will then be re-determined.

Table 1. Patch merging method

Patch merging method proposed	
<b>Input:</b> Plane parameters of initial planes;	<b>Output:</b> Plane parameters of final planes after merging;
<pre> 1: <b>for</b>(<math>i=0</math>; <math>i&lt;\text{Num\_plane}</math>; <math>i++</math>) {   <math>\text{numPt\_plane} \leftarrow 0</math>;   <b>for</b> (<math>j=0</math>; <math>j&lt;\text{TotalPts}</math>; <math>j++</math>)   {     <i>obtain the normal vector of the point and the normal vector of the plane:</i>     <math>\text{NV\_CurrentPt} = (\text{CurrentPt}.n_x, \text{CurrentPt}.n_y, \text{CurrentPt}.n_z)</math>; <math>\text{NV\_Plane} = (\text{Plane}.n_x, \text{Plane}.n_y, \text{Plane}.n_z)</math>;      <i>Compute the intersection angle between current point and the plane</i>     <math>\text{NormalAngle} = \text{NV\_CurrentPt} \cdot \text{NV\_Plane}</math>;      <i>Compute the distance from current point to the plane</i>     <math>d_{\text{point}} = -\text{NV\_CurrentPt} \cdot \text{Coordinate\_CurrentPt}</math>; <math>\text{Dist\_CurrentPt\_Plane} =  d_{\text{point}} - d_{\text{plane}} </math>;      2: <b>if</b> <math>\text{NormalAngle} &gt; \text{AngleThreshold}</math> &amp; <math>\text{Dist\_CurrentPt\_Plane} &lt; \text{DistThreshold}</math>         <i>Save the coordinate of current point;</i>         <math>\text{numPt\_plane}++</math>;     }   } }</pre>	



## (2) Re-judgment of coplanar points

There are two types of coplanar points. One is the intersection of two planes, which is the real coplanar point, also called the boundary point. The other is called the pseudo coplanar point, which actually belongs to one plane, but from a certain angle of view, it is a type of illusion due to the overlap of two or more planes.

Due to the normal vectors of coplanar points being so loose and disordered, the misclassification of coplanar points is related to the normal vector. Therefore, normal vector information must be considered in the re-judge process. They can be judged according to the distance from point to plane. The equation for the distance of any point  $P(x_p, y_p, z_p)$  to plane A is defined as:

$$D_A = |n_x x_p + n_y y_p + n_z z_p - (n_x x_0 + n_y y_0 + n_z z_0)| \quad (2)$$

In Equation (2),  $\{n_x, n_y, n_z\}$  represents the unit normal vector for plane A,  $(x_0, y_0, z_0)$  represents the central coordinate of the center of the plane. According to Equation (3),  $D_1, D_2, \dots, D_n$ , which are distances from the coplanar point C to N planes, are calculated; two minimum distances  $D_i, D_j \{i, j \in (1, n)\}$  are selected, and distance threshold between the point and the plane is set as  $\xi$ . If  $|D_i - D_j| \geq \xi$ , then the distance between the point and the two nearest planes is relatively large,  $|D_i| > |D_j|$  or  $|D_i| < |D_j|$ , which denotes that the point is a pseudo-coplanar point, belonging to the closer plane. If  $|D_i - D_j| < \xi$ , the distances of the point from the two closest planes are approximately equal, meaning that this point is a boundary point, and the neighborhood point auxiliary judge method is used to determine the boundary point.



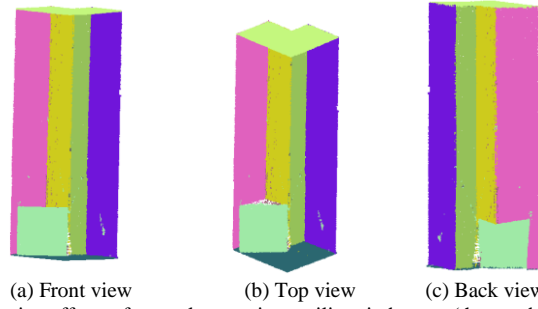


Figure 7. Plane extraction effects after coplanar point auxiliary judgment (the number of planes is 7)

When judging the neighborhood points, the neighborhood search is conducted by taking the pending point as the center of the circle and  $R$  as the radius to obtain the neighborhood point set. Then, we count the number of neighborhood points in each plane according to *plane\_flag*, a flag whose plane the pending point belongs to. Next, the point is classified into the plane with the maximum number of points, and the *plane\_flag* is reset for the pending point. Figure 7 shows the final extraction effect after auxiliary judgment of coplanar points. It can be seen that the pseudo plane points and boundary points are correctly classified to the corresponding planes, with 75417 out of a total of 75461 points being detected and achieving a recall of 99.94%.

### (3) Distance threshold

According to the above section, the value of distance threshold  $\xi$  directly affects the experimental results. In this section, we will analyze what the value of  $\xi$  can distinguish between pseudo coplanar points and boundary points. Taking *Sd1* for example, the pending point is regarded as the pseudo coplanar point while the distance difference between it and the two closest planes is bigger (at least 2 times). Usually,  $|D_i - D_j| > 0.001$ , as listed in Table 2. While the distance from it to the two closest planes is approximately equal, the pending point is a boundary point. Usually,  $|D_i - D_j| < 0.001$ , as listed in Table 3. Thus, we set  $\xi = 0.001$ . The plane extracting effect verified the feasibility of  $\xi$  value, as shown in Figure 8.

Table 2. Examples of pseudo coplanar points

No.	$D_i$	$D_j$	$ D_i - D_j $
1	0.0082	0.0016	0.0065
2	0.0072	0.0034	0.0037
3	0.0076	0.0038	0.0038
4	0.0098	0.0040	0.0057
5	0.0052	0.0021	0.0031
6	0.0376	0.0043	0.033
7	0.0166	0.0003	0.0163
8	0.0047	0.0018	0.0029
9	0.0249	0.0001	0.0247
10	0.0096	0.0015	0.0081

Table 3. Examples of boundary points

No.	$D_i$	$D_j$	$ D_i - D_j $
1	0.0040	0.0031	0.0009
2	0.0035	0.0028	0.0007
3	0.0028	0.0027	0.0001
4	0.0055	0.0049	0.0006
5	0.0031	0.0029	0.0002
6	0.0023	0.0019	0.0004
7	0.0053	0.0052	0.0001
8	0.0070	0.0069	0.0001
9	0.0077	0.0076	0.0001
10	0.0075	0.0069	0.0006

## 3. Experimental results and analysis

### 3.1. Testing data

To verify the feasibility and effectiveness of the mixed algorithm proposed in this study, building facade point clouds from three different scenarios were taken as testing data. The size of the data collected was from small to large, as shown in Table 4, in which *Sd1* (Sample data 1) is experimental data used in section 2. This section adopts the larger data volume samples *Sd2* and *Sd3*, and compares and analyses the proposed algorithm against classic algorithms, namely, RANSAC [11], region growing [10], dynamic clustering [10] and the standard Hough transform [8]. The CPU used in the experimental environment is an Intel Core i5 running at 2.6 GHz with 8 GB memory, and the hard disk is an 890-GB Kingston ATA unit.

Table 4. Testing data specification

Dataset	Number of points	Comments
<i>Sd1</i>	75,461	point clouds of a lab corner
<i>Sd2</i>	392,284	point clouds of a Research institute, in Zhengzhou
<i>Sd3</i>	1,189,222	point clouds of a Hotel, in Zhengzhou

### 3.2. Sensitivity analysis of thresholds

In this section, there are two important thresholds: angle threshold (*AngleThreshod*) and distance threshold (*DistThreshold*), which are used in every section of our proposed algorithm. In the section of FC and GHT, they adopt the same value. Here, we mainly discuss how they are determined and show how the performance of the method varies in term of these two thresholds.

#### (1) Angle threshold

In section 3.3, for choosing the most appropriate value for one threshold, a fixed value is used for another one. Firstly, we set a fixed value for *DistThreshold* in the program, where *DistThreshold* = 0.5m. For obtaining the most appropriate value for *AngleThreshod*, we change it from 6 to 20 gradually, as shown in the upper figure in Figure 8. The recall changes very little, from 0.998794 to 0.999443. While *AngleThreshod* = 15, 18 or 20, the recall reaches the max value. While *AngleThreshod* = 10, the recall is 0.999258. The difference is only 0.01%. For the other four algorithms (RANSAC, region growing, dynamic clustering, and SHT), we find that they have the highest performance when *AngleThreshod* = 10. Thus, in the series of comparison experiments, we set all angle thresholds equal to 10, which makes it possible to objectively compare the performance of the algorithms.

#### (2) Distance threshold

As with the angle threshold, for choosing the most appropriate value for distance threshold, we set a fixed value for *AngleThreshod* = 15 in the program. For obtaining the most appropriate value for *DistThreshold*, we enlarge it from 0.01 to 0.5 gradually, as shown in the lower figure in Figure 8. The recall also changes very little, from 0.998078 to 0.999443. While *DistThreshold* = 0.3, 0.4 and 0.5, the recall reaches the max value. For the other comparison algorithms, experimental results show that they reach the highest performance when *DistThreshold* = 0.5. Thus, in the series of comparison experiments, we set all distance thresholds equal to 0.5. Through the above analysis, we find that the recall changes very little when the two thresholds vary, which provides insight into the robustness of our proposed algorithm.

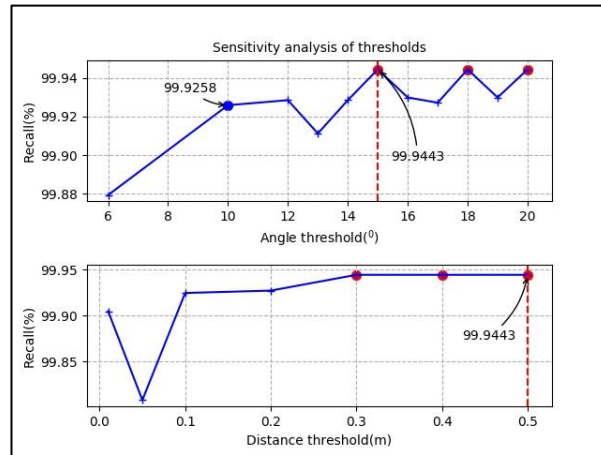


Figure 8. Sensitivity analysis of angle threshold and distance threshold

### 3.3. Algorithm comparison and analysis

To evaluate the performance of the mixed algorithm, four evaluation indicators are applied to conduct a quantitative analysis of the building facade features detected: numbers of planes extracted, detection time, numbers of accurate building facade points detected, and recall. Because the data used has been filtered, precision ratio is not considered. The recall is computed as

$$Recall = \frac{\text{numbers of relevant points that are retrieved}}{\text{the total numbers of relevant points in testing data}} * 100\% = \frac{RR}{RR + RNR} * 100\% \quad (3)$$

In Equation (3), RR (Relevant and Retrieved) indicates that points are relevant and they are retrieved. RNR (Relevant but Not Retrieved) indicates that points are relevant but they are not retrieved.



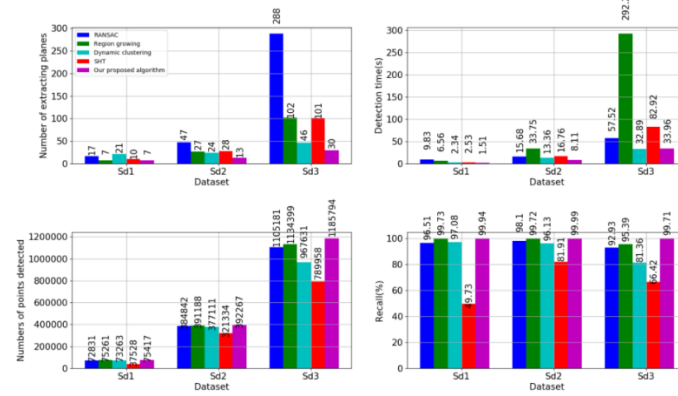


Figure 9. Comparison experiment results (four evaluation indicators)

Figure 9 shows the comparison experiment results between four classical algorithms with our proposed algorithm, through three different datasets. It is important to note that all experiments were conducted with the same thresholds, where  $AngleThreshold = 10$  and  $DistThreshold = 0.5$ . The upper left subfigure shows the numbers of extracting planes. On dataset Sd1, region growing algorithm extracts 7 planes, which is the same as our proposed algorithm, but consumes longer time. The larger the dataset, the longer it takes. On dataset Sd3, consumed time increases to 292.29s, which is the longest. The number of planes extracted by the RANSAC algorithm is far greater than that of the actual facades on Sd3, as the over-segmentation phenomenon occurs. The upper right subfigure shows the detection time, namely, searching speed. The dynamic clustering algorithm has the fastest speed aside from our proposed algorithm, but fewer points are classified into the correct planes and the recall is lower, as shown in the lower left subfigure and the lower right subfigure. For RANSAC and SHT, the numbers of extracting planes are far more than the real numbers and have longer detection times; additionally, the recall of SHT is the smallest. Through experimental comparison and analysis of the four evaluation indicators, the proposed algorithm has the best performance: the fastest speed and the highest recall. The plane extraction effects are shown in Figure 10 and Figure 11.

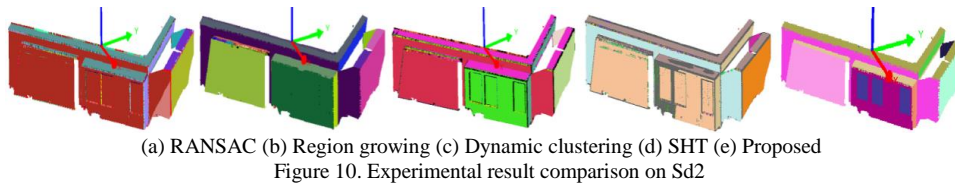


Figure 10. Experimental result comparison on Sd2

As shown in Figure 10, the proposed algorithm is superior to the traditional region growing, dynamic clustering, and SHT algorithm. Building walls, windows, balconies, doors, and other prominent or concave parts are extracted effectively.

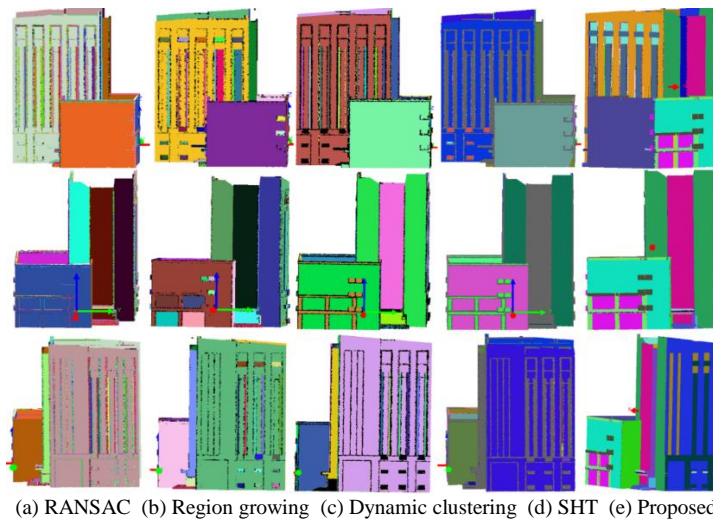


Figure 11. Experimental result comparison on Sd3 (The first row indicates the extracting effect from left view, the second row indicates front view, and the third row indicates the right view)

As shown in Figure 11, the boundary points in graph (a), (b), (c) and (d) are not classified into the correct planes. Graph (e) provides the best extraction effect, and there is no insufficient-segmentation or over-segmentation phenomena. Moreover, all boundary points and coplanar points are affiliated with the corresponding planes, and the more points in the building point cloud, the more obvious the effect.

#### 4. Conclusions

Regarding building facade feature extraction based on 3D laser point clouds, this study analyses the advantages and disadvantages of existing methods. To conduct semantic segmentation for building facades rapidly and effectively, this study first establishes a Kd-OcTree mixed index to organize and manage point clouds to prepare them for faster retrieval. The normal vectors of each point are calculated, and fuzzy clustering is carried out in each partition to obtain clusters. Then, FC-GHT transformation is performed for each cluster to extract preliminary facades. Computation quantity and memory consumption reductions are considered from multiple angles during the entire process. After extracting the preliminary facades, similar patches are merged according to plane distances and plane normal vectors, and coplanar points are reclassified according to the distance threshold  $\xi$  and neighborhood point auxiliary judgment method. Finally, the accurate facades are extracted. The results show that our proposed algorithm is effective and has the best effect and fastest speed among all of the algorithms tested. The included angle and distance of normal vectors are the two main thresholds, and we have analyzed how they are determined and how the performance of the proposed algorithm varies in terms of the thresholds. Compared to traditional algorithms, the automation of the proposed algorithm has been improved, and the extraction effect achieves the expectation.

#### Acknowledgements

This study was undertaken with financial support from the National Natural Science Foundation of China (NSFC) (Grant no. 41501491 and 61601184) and the Key Science Research Program of Higher Education of Henan Province, China (No. 16A520062).

#### References

1. X. Y. Ai and L. Y. Wang, "Extraction of Planar Characteristics of Airborne LiDAR Point Cloud Data". *Journal of Liaoning Technical University: Natural Science*, 2015, 34(2): 212-216.
2. N. H. Arachchige, S. N. Perera, and H. G. Maas, "Automatic Processing of Mobile Laser Scanner Point Clouds for Building Facade Detection". *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXIX-B5, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia
3. J. M. Biosca and J. L. Lerma, "Unsupervised Robust Planar Segmentation of Terrestrial Laser Scanner Point Clouds based on Fuzzy Clustering Method". *ISPRS Journal of Photogrammetry & Remote Sensing* 2008, 63: 84-98
4. J. P. Burochin, B. Vallet, M. Bredif, et al. "Detecting Blind Building Facades from Highly Overlapping Wide Angle Aerial Imagery". *ISPRS Journal of Photogrammetry and Remote Sensing*. 96: 193-209, 2014
5. Y. C. Feng, "The Quickly Automated Extraction of Building Facades Information from Mobile LiDAR Point Clouds". Thesis (PhD). Southwest Jiaotong University, 2014.
6. R. Krishnapuram and D. Casasent, "Determination of Three-dimensional Object Location and Orientation from Range Images". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1989, 11(11): 1158-1167.
7. M. D. Li, S. P. Jiang, and H. P. Wang, "A RANSCA-based Stable Plane Fitting Method of Point Clouds". *Science of Surveying and Mapping*. 40(1), 2015: 102-106.
8. M. L. Li, G. Y. Li, L. Wang, et al. "Automatic Feature Detecting from Point Clouds Using 3D Hough Transform". *Bulletin of Surveying and Mapping*, 2015(2): 29-33.
9. M. L. Li, R. Zhang, and G. Y. Li. "Accurate Normal Calculating and Surface Smoothing of Laser-Scanned Point Clouds". *Journal of Computer-Aided Design & Computer Graphics*. 2015, 27(7): 1153-1161
10. M. L. Li, "Technology of Preprocessing on 3D Laser Scanned Point Clouds". Master D. Dissertation of PLA Information Engineering University, 2014.
11. N. Li, Y. W. Ma, Y. Tang, S. L. Gao, "Segmentation of building Façade Point Cloud using RANSAC". *Science of Surveying and Mapping*. 36(5): 144-146. 2011
12. Y. N. Lin and W. Wei, "Research on Algorithm of Object Tracking based on Generalized Hough Transform". *ZheJiang University*, ph.D, 2013.
13. H. A. Nalani and H. G. Maas, "Automatic Building Facade detection in Mobile Laser Scanner Point Clouds". In: *The German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF)*, Potsdam, Germany. 2012. (CD-ROM).
14. V. Petar, K. Igor, R. S. Marko, "Obtaining Structural Descriptions of Building Facades". *Computer Science and Information Systems* 13(1): 23-43, 2015
15. X. Y. Song, S. Yuan, H. B. Guo, et al., "Pattern Identification Algorithm with Adaptive Threshold Interval based Extended Hough Transform". *Chinese Journal of Scientific Instrument*. 2014, 35(5): 1109-1117.
16. G. Vosselman, B. G. Gorte, G. Sithole, et al. "Recognising Structure in Laser Scanner Point Cloud", in *International Archives*

- of Photogrammetry[C]. Remote Sensing and Spatial Information Sciences, 2004.
17. Z. Wei, "Automated Extraction of Buildings and Facades Reconstruction from Mobile LiDAR Point Clouds". Wuhan University, 2012.
  18. Z. Wei, B. S. Yang, and Q. Q. Li, "Automated Extraction of Building Footprints from Mobile LIDAR Point Clouds". Journal of Remote Sensing, 16(2): 286–296, 2012.
  19. L. Yan, H. Xie, X. B. Hu, X. W. Bao, "A New Hybrid Plane Segmentation Approach of Point Cloud". GEOMATICS AND INFORMATION SCIENCE OF WUHAN UNIVERS, 2013, 38(5): 517-521.
  20. B. S. Yang, Z. Dong, Z. Wei, et al., "Extracting Complex Building Facades from Mobile Laser Scanning Data". Acta Geodaetica et Cartographica Sinica, 42(3):411-417, 2013.
  21. B. S. Yang, Z. Wei, Q. Q. Li, J. Li, "Automated Extraction of Street-Scene Objects from Mobile Lidar Point Clouds", International Journal of Remote Sensing, 33(18):5839-5861,2012.
  22. B. S. Yang, Z. Wei, Q. Q. Li, J. Li, "Semiautomated Building Facade Footprint Extraction From Mobile LiDAR Point Clouds". IEEE Geoscience and Remote Sensing Letters,10(4): 766-770, 2013
  23. D. Y. Zhang, W. Q. Wu, M. P. Wu, et al., "Plane Landmark Detection from Lidar Data Based on 3D Hough Transform". JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY, 2010,32(2):130-134.
  24. R. Zhang, G.Y. Li, L. Wang, M.L. Li, Y.L. Zhou, "New Method of Hybrid Index for Mobile LiDAR Point Cloud Data". Geomatics and Information Science of Wuhan University. DOI: 10.13203/j.whugis20160441, 2017.
  25. R. Zhang, G. Y. Li, M. L. Li, et al. "Classification of LiDAR Point Clouds based on PCA-BP Algorithm". Bulletin of Surveying and Mapping, 2014(7): 23-26.
  26. F. Zhou, C. Yang, C. G. Wang, et al., "Circle Detection and its Number Identification in Complex Condition based on Random Hough Transform". Chinese Journal of Scientific Instrument. 2013, 34(3):622-628.

**Rui Zhang** received her M.Sc. degree from Southwest University in 2006. She is currently a Ph.D. candidate at Information Engineering University. She works at the North China University of Water Resources and Electric Power as an associate professor. Her main research interests include data processing of point clouds, computer vision, and artificial intelligence.

**Jiayi Wu** received her M.Sc. degree from Warwick University, with a major in E-business. She works at the North China University of Water Resources and Electric Power as a teaching assistant. Her research direction is software engineering.

**Guangyun Li** received his M.S. degree in engineering in 1987 from the Zhengzhou Institute of Surveying and Mapping. Currently, he works at the Information Engineering University as a professor and doctoral supervisor. His main research interests include precise engineering and industry measurement as well as navigation and location services and applications.

**Li Wang** received his Ph.D. degree in engineering in 2014 from the Information Engineering University. He currently works at the Information Engineering University. His main research interests include precise engineering and industry measurement.