

## Article

# Building Point Detection from Vehicle-Borne LiDAR Data Based on Voxel Group and Horizontal Hollow Analysis

Yu Wang <sup>1,2</sup>, Liang Cheng <sup>1,2,3,4,\*</sup>, Yanming Chen <sup>1,2</sup>, Yang Wu <sup>1,2</sup> and Manchun Li <sup>1,2,3,4,\*</sup>

<sup>1</sup> Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Nanjing University, Nanjing 210093, China; wangyu4458@gmail.com (Y.W.); aiorischan@gmail.com (Y.C.); Wuyang\_nju@126.com (Y.W.)

<sup>2</sup> Department of Geographic Information Science, Nanjing University, Nanjing 210093, China

<sup>3</sup> Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Nanjing 210093, China

<sup>4</sup> Collaborative Innovation Center for the South Sea Studies, Nanjing University, Nanjing 210093, China

\* Correspondence: lcheng@nju.edu.cn (L.C.); limanchun\_nju@nju.edu.cn (M.L.); Tel.: +86-25-8359-7359 (L.C.); +86-25-8359-7359 (M.L.)

Academic Editors: Randolph H. Wynne and Prasad S. Thenkabail

Received: 11 March 2016; Accepted: 20 April 2016; Published: 17 May 2016

**Abstract:** Information extraction and three-dimensional (3D) reconstruction of buildings using the vehicle-borne laser scanning (VLS) system is significant for many applications. Extracting LiDAR points, from VLS, belonging to various types of building in large-scale complex urban environments still retains some problems. In this paper, a new technical framework for automatic and efficient building point extraction is proposed, including three main steps: (1) voxel group-based shape recognition; (2) category-oriented merging; and (3) building point identification by horizontal hollow ratio analysis. This article proposes a concept of “voxel group” based on the voxelization of VLS points: each voxel group is composed of several voxels that belong to one single real-world object. Then the shapes of point clouds in each voxel group are recognized and this shape information is utilized to merge voxel group. This article puts forward a characteristic nature of vehicle-borne LiDAR building points, called “horizontal hollow ratio”, for efficient extraction. Experiments are analyzed from two aspects: (1) building-based evaluation for overall experimental area; and (2) point-based evaluation for individual building using the completeness and correctness. The experimental results indicate that the proposed framework is effective for the extraction of LiDAR points belonging to various types of buildings in large-scale complex urban environments.

**Keywords:** vehicle-borne LiDAR; building point extraction; voxel group; horizontal hollow analysis

## 1. Introduction

With the rise of the “smart city” concept, automatic information extraction and three-dimensional (3D) reconstruction of buildings in urban areas have become popular research topics in the fields of photogrammetry and remote sensing, computer vision, *etc.* [1]. The results of related studies have been widely used in fields such as intelligent building [2], virtual roaming [3], ancient architecture conservation [4], disaster management [5], and others. A laser scanning system has the unique advantage of providing data of directly measured 3D points in building extraction and 3D reconstruction [6]. According to its carrying platform, a laser scanning system can be divided into satellite-based laser scanning, airborne laser scanning (ALS), vehicle-borne laser scanning (VLS), and terrestrial laser scanning (TLS) systems [7]. Among these types, ALS has received considerable attention for applications in urban regions [8], and various methods have been proposed for 3D

building modeling [9–15], change detection [16,17] and information extraction in roads [18,19], parking lots [20], buildings [21–23] and land cover [24]. Moreover, the fusion of ALS data and optical image expands the scope of application and makes up some defects of the ALS data [25].

Compared to ALS, VLS can scan the facades of buildings and obtain denser point clouds with higher accuracy [18]. VLS is therefore suitable for 3D reconstruction of building facades. However, highly dense point clouds, with a huge amount of data, means more redundancies and occlusions. Strong variations of point densities occur on the building's surface, which are caused by the viewpoint limitation of the scanner [26,27]. Due to these factors, it is still a technical challenge to accurately and quickly extract building point information from VLS data.

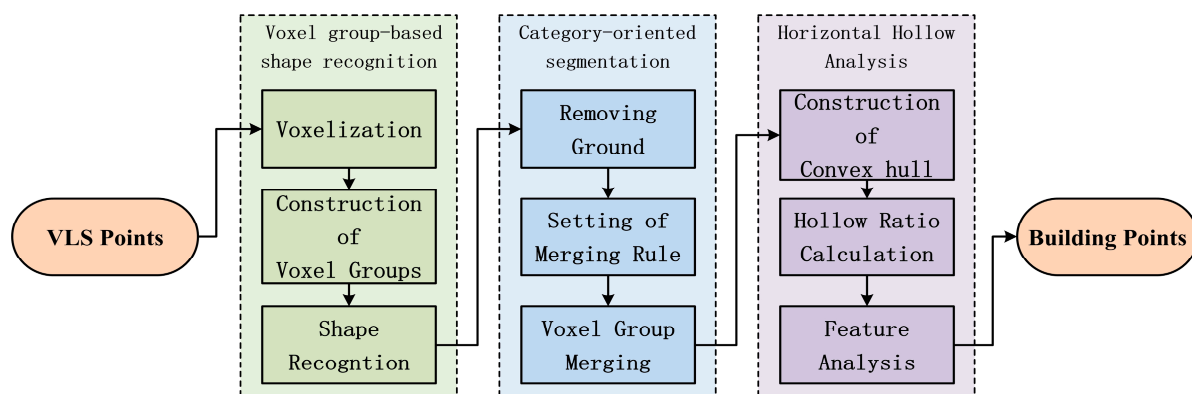
Some scholars employ a bottom-up approach for building information extraction from VLS data. Manandhar *et al.* [28], for example, segmented point clouds into separate scan lines to distinguish artificial and natural objects, according to the differences of geometric properties or spatial distribution of the points on each scan line. The scan lines were then classified as artificial objects, which were combined by prior knowledge for extracting buildings. A few scholars have aimed to identify specific geometries from point clouds—typically flat or curved surfaces—as a way to extract buildings. Common methods have included the random sample consensus (RANSAC) algorithm and Hough transform [29–32]. Moosmann *et al.* [33] proposed a region-growing method based on local curvature criterion to quickly divide points into plane. Munoz *et al.* [34] classified point clouds as pole, scatter or facade using the associative Markov network. Pu *et al.* [18] proposed a knowledge-based feature recognition method to recognize planes and poles from point clouds as basic structures for extracting building facades or street lamps. However, the above methods did not consider that the strong variation of the point densities will cause significant changes of accuracy. In order to solve this problem, Demantké *et al.* [35] presented an approach for recognizing shape of each point based on Principal Component Analysis (PCA) and adopted two radius-selection criteria called entropy feature and the similarity index for choosing the best neighborhood scale. Similarly, a procedure was introduced by Yang *et al.* [36] that include Support Vector Machine (SVM), which can divide point clouds into three main classes: linear, planar, and spherical. On this basis, Yang tried to identify complex building facades and other objects with semantic knowledge [37].

In addition, some scholars have adopted a top-down approach to extracting building information from mobile LiDAR data. Li *et al.* [38] for example, created point clouds projected onto a two-dimensional (2D) plane grid; according to the density and other information, they then extracted the building contour. Aijazi *et al.* [39] proposed a super-voxel-based approach to segment discrete point clouds. They additionally suggested use of the link-chain method to merge super-voxels into individual objects; the point clouds are then classified as buildings, roads, pole-like objects, vehicles, or trees according to the direction of the surface normal, reflection intensity, and geometry properties. Furthermore, Yang *et al.* [40] generated a geo-referenced feature image from point clouds, they then adopted discrete discriminant analysis to segment point clouds into separate objects for extracting buildings and trees. This method was likewise used to extract building footprints [1]. Among these methods, high precision building extraction results often depends on the points cloud segmentation results, which are always difficult to be obtained in extremely complex scene.

Despite that, many related researches have been reported, and further studies are still urgent as to how to automatically and efficiently extract various types of building points [41], especially in large-scale complex environments of urban regions, such as skyscrapers, low cottages, ordinary residences, or stadiums. Different types of urban areas, such as commercial and residential districts, require different semantic rules, or parameters and thresholds, which are heavily dependent on user experience. In addition, for most of these methods, each LiDAR point involved in the calculation is to increase the processing time.

To address these problems, this paper proposes an approach on automatically extracting various types of buildings, from vehicle-borne laser scanning data acquired from the large-scale complex environments of urban regions (see Figure 1). First, a new structure, called “voxel group”, is applied to

further organize voxels based on the voxelization of VLS points. The method will process each voxel group instead of discrete point clouds to accelerate the calculation. Then a simple way is applied to quickly recognize the shape of point clouds in each voxel group. Second, a category-oriented merging is used for merging voxel group by utilizing the shape information to obtain high precision point clouds clusters. At last, a novel characteristic nature of vehicle-borne LiDAR building points, called “horizontal hollow ratio” for efficient extraction of various forms of buildings.



**Figure 1.** Flowchart of building point extraction from VLS point data.

## 2. Methods

### 2.1. Voxel Group-Based Shape Recognition

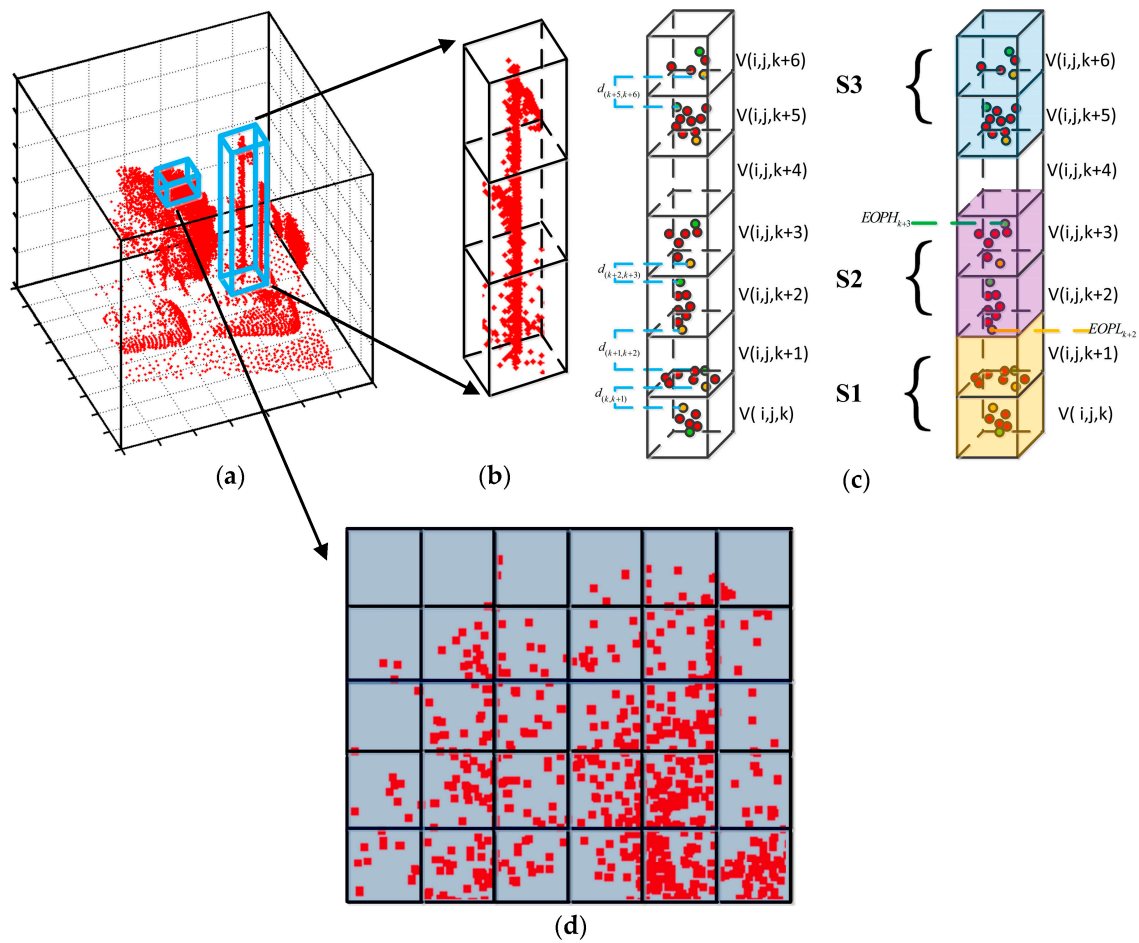
#### 2.1.1. Voxelization

The voxel is represented by a standard cube that records the serial number of the contained LiDAR points. A simple but efficient way to segment point clouds, voxelization has been widely used in the field of forest science. For example, some scholars have segmented the ALS or TLS data with voxels to analyze forest structure [42,43] or to describe the morphological structures of tree canopies [44], determine crown-base height [45], and separate point clouds into individual trees [46]. Wu *et al.* [3] used a voxel-based method to extract street-lining trees from VLS LiDAR data. Wang *et al.* [47] generated DEM using a voxel-based method. Jwa *et al.* [48] automatically extracted power lines with a method based on voxelization. Owing to its simplicity and ease of representation both visually and in data structures [49], voxel-based methods have been suitable for model reconstruction by LiDAR data. Park *et al.* [50], Hosoi *et al.* [51], and Stoker [52] adopted voxel-based methods to build 3D models of individual trees, Cheng using voxel-based methods for reconstruction of large multilayer interchange bridge [53] and urban power line [54] while some scholars have applied it for building-model reconstruction with several good results [44–47].

#### 2.1.2. Generating of Voxel Group

Voxel-based methods can transform the extraction of disorderly distributions of discrete points into the filtering of voxels with topological relations [54]. This approach is suitable for vehicle LiDAR points. However, when dealing with large-scale urban LiDAR data, the large volume of voxels remains an obstacle for rapid process. Therefore a new structure called the “voxel group” has been put forward to further organize voxels; each voxel group is composed of several voxels that are adjacent to each other and have the same geometric properties, as shown in Figure 2. Voxel group construction is based on the following two assumptions: (1) A series of voxels with the same horizontal coordinates, with elevations adjacent to each other, have a greater possibility of belonging to the same single real-world object, as depicted in Figure 2b; (2) One voxel and the adjacent voxels with small elevation differences are more likely belong to the same object, as shown in Figure 3d. The above two hypotheses are derived

from the respective facts that the object's structure has continuity in vertical directions, regardless whether it is an artificial or natural object.



**Figure 2.** Construction of voxel group. (a) Point clouds distribution of several objects in a 3D voxel grid system; (b) Street lamp point clouds and the generated voxels; This is a typical case in which the voxels with the same horizontal and vertical coordinates with adjacent elevations belong to the same target; (c) Schematic of the process of dividing the voxel distributions on the same vertical direction; (d) Profile of part canopy of a street tree, a case that adjacent voxel within points belong to one object have little elevation differences.

The process of establishing a voxel group is as below.

**Step 1: Building 3D voxel grid system.** Set an appropriate size  $S$  to build a regular 3-D voxel grid system. Each LiDAR point is added to each voxel according to its 3D coordinates. The minimum value among all LiDAR point coordinates ( $x_{\min}, y_{\min}, z_{\min}$ ) is the origin of the 3D voxel grid system. For each LiDAR point, the row, column, and layer number ( $i, j, k$ ) of its corresponding voxel are recorded to construct a two-way index.

**Step 2: Dividing voxels in each column (in vertical direction).** On account of a series of voxels distributed in the same vertical direction, the group of voxels with the same row and column ( $i, j$ ) may belong to a different target, such as pedestrians or vehicles below the canopy of trees along a street. Therefore, these voxels must be separated to ensure that each voxel group contains only one object's points, as shown in Figure 2c. Accordingly, the elevation difference between voxel  $V(i,j,k)$  and the voxel above it,  $V(i,j,k+1)$  should be calculated:

$$D_{(k,k+1)} = EOP_{k+1} - EOP_k \quad (1)$$



$EOP_{k+1}$  indicates the maximum value of the LiDAR point elevation contained within voxel  $V_{(i,j,k+1)}$ , and  $EOP_k$  represents the maximum value of the LiDAR point elevation contained within voxel  $V_{(i,j,k)}$ . Threshold  $T_s$  is to be set and if  $D_{(k,k+1)} \leq T_s$ , then  $V_{(i,j,k+1)}$  will join the voxel column, including  $V_{(i,j,k)}$ ,  $S = \{\dots v_{(i,j,n)}, \dots v_{(i,j,k)}\}$ , to form a new voxel column,  $S = \{\dots v_{(i,j,n)}, \dots v_{(i,j,k)}, v_{(i,j,k+1)}\}$ .

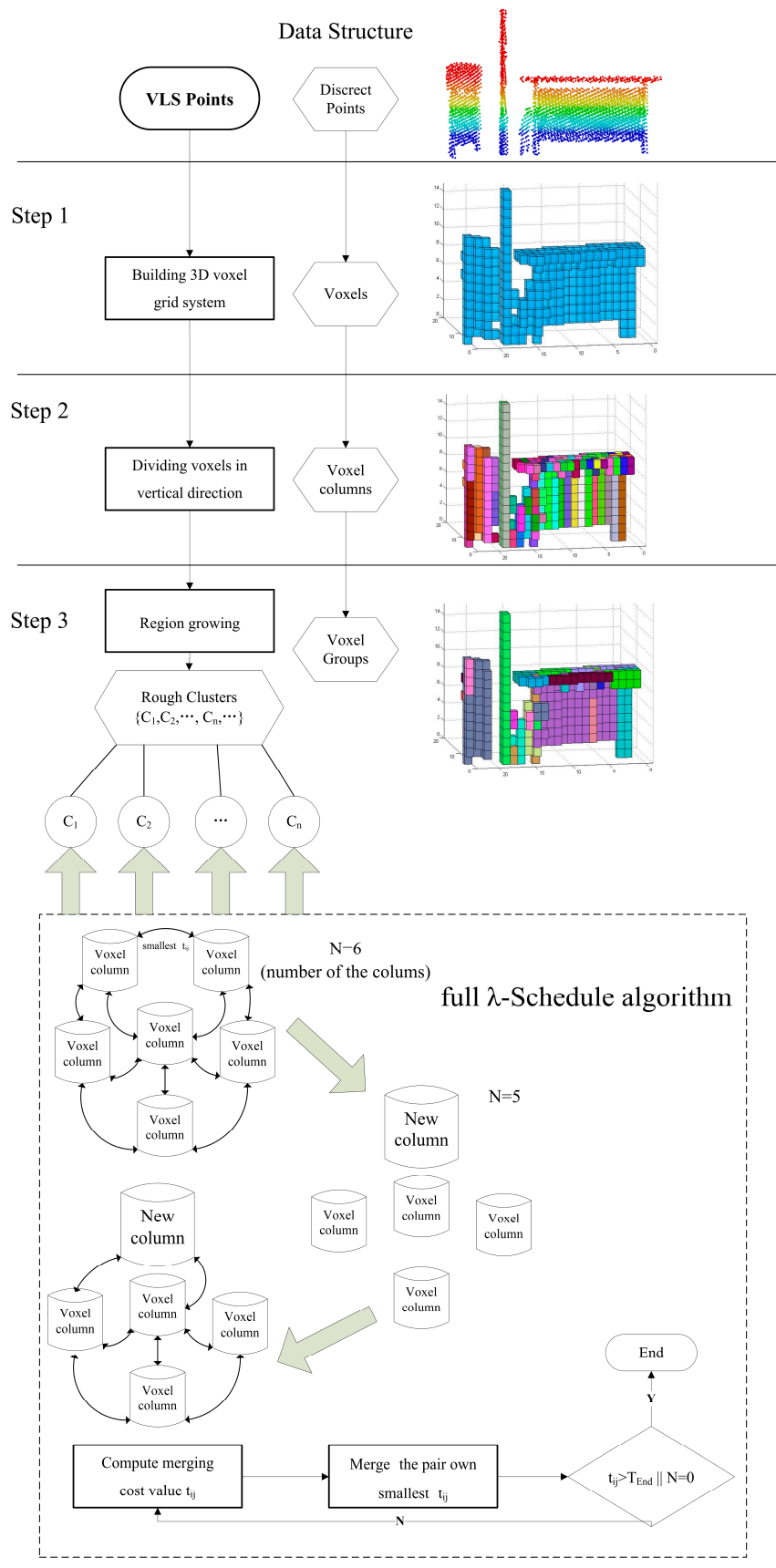
Step 3: **Merging process in horizontal direction for voxel group.** A full  $\lambda$ -Schedule algorithm [55] is to be taken to merge the voxel columns in horizontal direction. The full  $\lambda$ -Schedule algorithm [55] was first used to segment SAR images. The segmentation principle is based on the Mumford–Shah energy equation to judge the difference in object attributes and the complexity of the object boundary [11]. The merging cost value  $t_{i,j}$  of each adjacent voxel column  $(S_i, S_j)$  is calculated as below:

$$t_{i,j} = \frac{\frac{|S_i^A| \cdot |S_j^A|}{|S_i^A| + |S_j^A|} ||S_i^E - S_j^E||}{\ell(\partial(S_i, S_j))} \quad (2)$$

$(S_i, S_j)$  are two adjacent voxel columns in horizontal direction.  $S_i^A$  is the horizontal projection area of voxel column, which is calculated by the defined length  $l$ , width  $w$  and the number of the horizontal projection grids.  $S_i^E$  is the elevation value of the highest LiDAR point within voxel column.  $\ell(\partial(S_i, S_j))$  is the length of the shared boundary consists of two parts: the length in vertical direction and the length in horizontal direction. The details are as below:

- i Take a simple region growth for whole voxel columns in horizontal direction based on connectivity to get several rough clusters:  $\{C_1, C_2, \dots, C_n, \dots\}$ .
- ii Compute all the pairs of adjacent voxel columns within  $C_n$  and their merging cost value from Equation (6) and sort them into a list.
- iii Merge the pair  $(S_i, S_j)$  which own smallest  $t_{i,j}$  to form a new voxel column  $S_{ij}$  and update the merging cost value.
- iv Repeat the step ii and step iii until the  $t_{i,j}$  exceeds the threshold  $T_{End}$  or all the voxel columns within  $C_n$  into one group.
- v Repeat the step ii, iii, iv until all clusters are processed.

The proposed method take a connectivity-based region growth as first in Step 3 is because the computational complexity of the full  $\lambda$ -Schedule algorithm is  $O(mg \log_2(mn))$  for a 2D image of  $m \times n$  pixels [56]. For 3D voxel grid system, the computational complexity will be higher so the origin 3D voxel grid system must be divided into pieces to reduce the amount of involved voxel columns in one process. Finally, all voxel columns are combined into a higher-level structure, the voxel group. The LiDAR points within each voxel group belong to the same single real-world object and have the same geometric properties or shape information.



**Figure 3.** Flowchart of generating of voxel groups.

### 2.1.3. Shape Recognition of Each Voxel Group

Demantke *et al.* [35] and Yang *et al.* [36] used a PCA-based method to identify the shapes of point clouds. They divided whole point clouds into three categories—linear, planar, and spherical and proved that fusion of point clouds shape information can efficiently segment mobile laser-scanning of point clouds of large-scale urban scenes into single objects. PCA is a common method for analyzing the spatial distributions of neighborhoods of points. It results in a set of positive eigenvalues:  $\lambda_1, \lambda_2, \lambda_3$ , ( $\lambda_1 > \lambda_2 > \lambda_3$ ). Then Demantke *et al.* [35] proposed the dimensionality features to describe linear ( $a_{1d}$ ), planar ( $a_{2d}$ ) and spherical ( $a_{3d}$ ) within.  $V_p^R$ .  $V_p^R$  represents the neighboring points of point  $p$  with the neighborhood size  $R$ .

$$a_{1d} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}} \quad (3)$$

$$a_{2d} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \quad (4)$$

$$a_{3d} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \quad (5)$$

A proper neighborhood size is the key for good shape recognition results. Demantke *et al.* [35] proposed an entropy function that equations the dimensionality of features derived from the eigenvalues of each point:

$$E_f(V_p^R) = -a_{1d}\ln(a_{1d}) - a_{2d}\ln(a_{2d}) - a_{3d}\ln(a_{3d}) \quad (6)$$

When  $E_f(V_p^R)$  achieve the minimum value,  $V_p^R$  have the most possibility belong to one dimensionality feature:

$$d(V_p^R) = \underset{n \in [1,3]}{\operatorname{argmax}}(a_{nd}) \quad (7)$$

To acquire the optimal neighborhood size of each point, an initial neighborhood size value must be set; it is then gradually increased until the entropy function attains the minimum value. Yang *et al.* [36] fused the intensity of each point in the process of determining the best neighborhood size to improve the accuracy of estimation of shape features. Both two methods required several calculations of eigenvectors and eigenvalues for each point when the neighborhood size changes; therefore, it is time-consuming and not suitable for handling large-scale urban LiDAR data, even though it can achieve a high accuracy.

Unlike the work of Demantke *et al.* [35] and Yang *et al.* [36], this article proposes a simple and rapid approach that takes advantage of the voxel group concept for estimation of shape features. The flowchart is shown in Figure 4. Each voxel group is taken as a unit to be estimated its shape. To speed up the process of shape estimation, a part of points in a voxel group is selected instead of its whole points. Because each voxel group may contain point clouds of one tiny single real-world object or a local part of one large single real-world object, the points within one voxel group should have same geometric properties or shape features. The detailed procedure for shape features identifying can be described as follows.

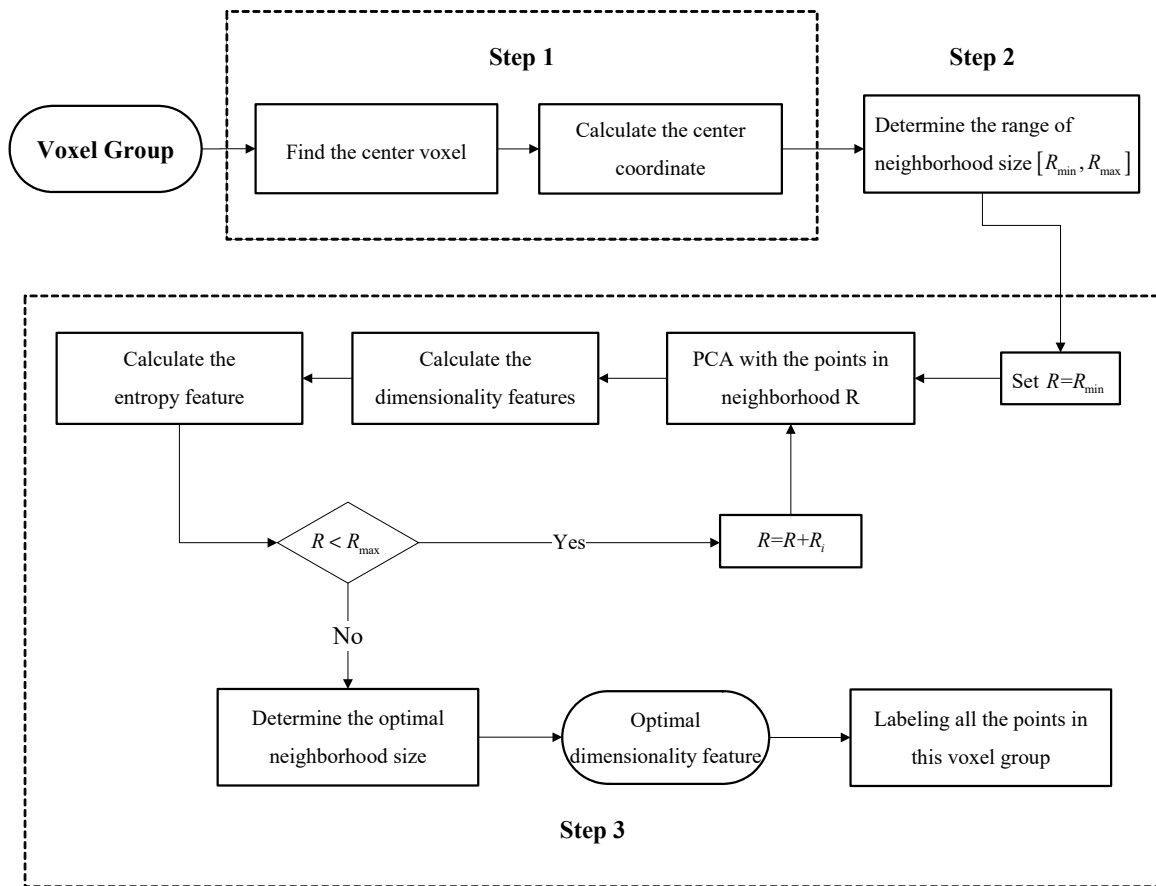


Figure 4. Flowchart of shape recognition for each voxel group.

Step 1: *Finding the center voxel*. The point density of each voxel within one voxel group is calculated and finds the most dense voxel  $V_{md}$ . Calculate the center coordinate of points in this voxel:

$$(\bar{X}, \bar{Y}, \bar{Z}) = \left( \frac{\sum_{i=0}^n x_i}{n}, \frac{\sum_{i=0}^n y_i}{n}, \frac{\sum_{i=0}^n z_i}{n} \right) \quad (8)$$

Step 2: *Determine the variation range of neighborhood size*. Centering on  $(\bar{X}, \bar{Y}, \bar{Z})$ , the minimum neighborhood size  $R_{min}$  is determined as the radius that includes the minimal number  $N_p$  of points required for PCA. Set the increment  $R_i$ , the neighborhood size will increase until the radius reach the boundary of voxel group. Then the variation range of neighborhood size  $[R_{min}, R_{max}]$  is obtained.

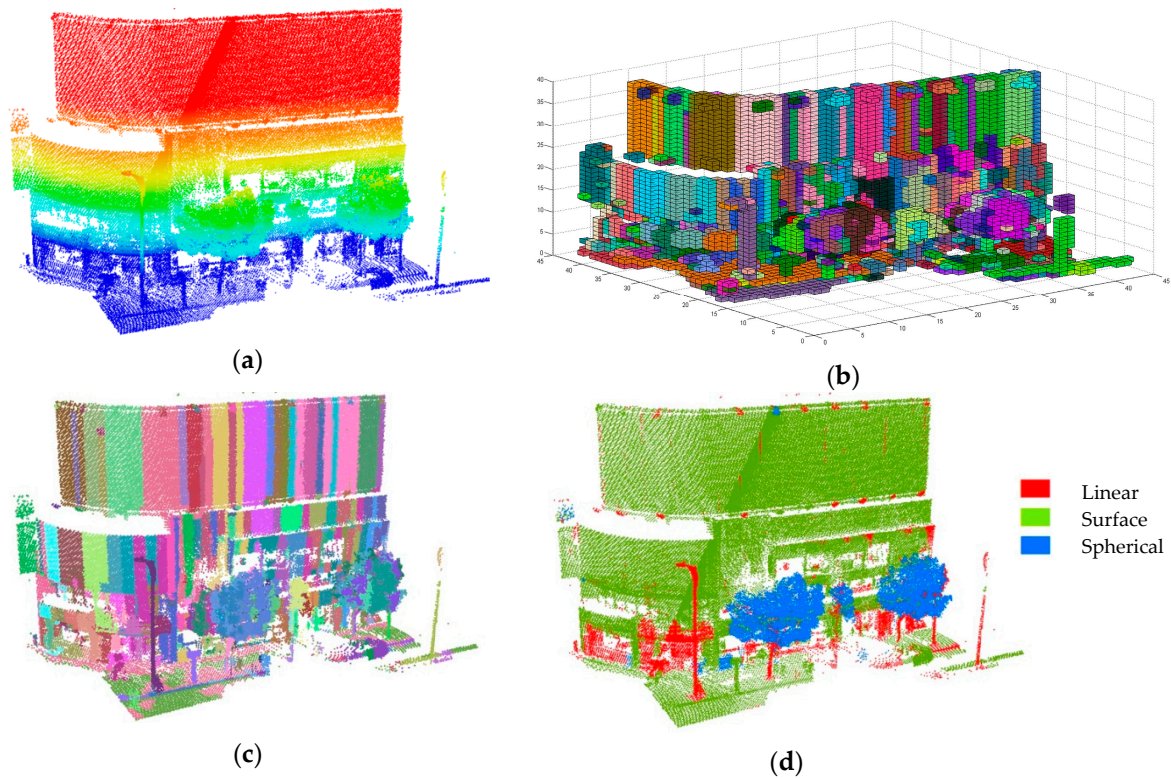
Step 3: *Calculate the dimensionality features and entropy feature*. Then the dimensionality features  $a_{1d}, a_{2d}, a_{3d}$  and entropy feature  $E_f(V_p^R)$  within  $V_p^R$  ( $R \in [R_{min}, R_{max}]$ ) are calculated by the Equation (9). In this paper,  $P$  denotes the center coordinate of points in the selected voxel. Then the optimal neighborhood size can be obtained:

$$R_{opt} = \underset{R \in [R_{min}, R_{max}]}{\operatorname{argmin}} (E_f(V_p^R)) \quad (9)$$

Then the dimensionality feature of  $V_p^{R_{opt}}$  can be identified by Equation (7). The voxel group and the whole points within it will be labeled the same dimensionality feature.

Every voxel group and point clouds within it are divided into three shape categories: linear, planar and spherical (see Figure 5d) by the Step 1–3 above. The principal direction of point clouds

in a linear voxel group, the surface normal direction of point clouds in a planar voxel group, and the coordinates of center point clouds in a spherical voxel group can be further obtained.



**Figure 5.** Voxel group-based shape recognition. (a) Raw LiDAR point clouds include building facades, street trees, street lamps, cars, and the ground; (b) Generated voxel group, voxels of the same color belong to the same voxel group; (c) LiDAR points within each voxel group, points of the same color belong to the same voxel group; (d) Shape recognition results.

## 2.2. Category-Oriented Merging

As noted, each voxel group and the point clouds within it are divided into one shape category: linear, planar, or spherical. The proposed method merges discrete voxel groups into a single real-world object on the ground fusion of the shape information for building detection. This process includes two steps: removing ground points and category-oriented merging.

### 2.2.1. Removing Ground Points

When addressing outdoor 3D data, an estimate of the ground plane provides an important contextual cue [57]. Particularly in large-scale urban regions, advance ground point removal can greatly reduce the amount of data and improve efficiency. A simple strategy is used to quickly filter ground points based on the establishment of the voxel group and shape recognition. The steps of this strategy are outlined below.

**Step 1: Extracting the potential voxel group that contains ground points.** The difference value between the lowest and highest points is calculated for each planar voxel group with an angle between the surface normal vector and horizontal plane that is greater than  $85^\circ$ :

$$D_{\max} = P_{h\max} - P_{h\min} \quad (10)$$

$P_{h\max}$  is the elevation value of the highest point and  $P_{h\min}$  is the elevation value of the lowest point.



**Step 2: Combining the connected region.** If the elevation difference between two adjacent candidate voxel groups contains ground points less than 0.3 m, then the two voxel groups are merged. Repeat this process and calculate the area of the final combined voxel group:

$$Area = N \times S^2 \quad (11)$$

where  $N$  is the number of voxels within the combined voxel group.

**Step 3: Removing ground points.** Set the area threshold 10 m<sup>2</sup> to filter the too small combined voxel group. Then all candidate voxel groups' average elevation are recorded and the outliers are rejected, always indicating the suspended flat roof. All the points within the rest of candidate voxel groups will be labeled as ground points and need to be removed before the next step.

### 2.2.2. Category-Oriented Merging

A single real-world object may consist of more than one shape, such as a building composed of a planar roof and linear columns. Therefore, different rules are set for merging adjacent voxel groups containing non-ground points that have the same or different shape properties based on a region growing algorithm.

There come several combinations according to the type of the candidate voxel group: (1) two linear voxel group; (2) two planar voxel group; (3) two spherical voxel group; (4) one linear and one planar voxel group; (5) one linear and one spherical voxel group. Generally, components of artificial objects are approximate parallel or perpendicular to each other, such as palisade tissue, billboard and its stanchion, traffic sign's cross arm and upright, building's flashing and facade. When dealing with the combination 1, 2, 4, the principal direction differences or normal vector differences between the two linear voxel groups or two planar voxel groups and the angle differences between the principal direction of the linear voxel group and the normal vector of the planar voxel group firstly. Two parallel candidate voxel groups (differences smaller than 10°) will be required corresponding judging conditions different from the two candidate voxel groups perpendicular to each other. The merging rules are shown in Table 1.

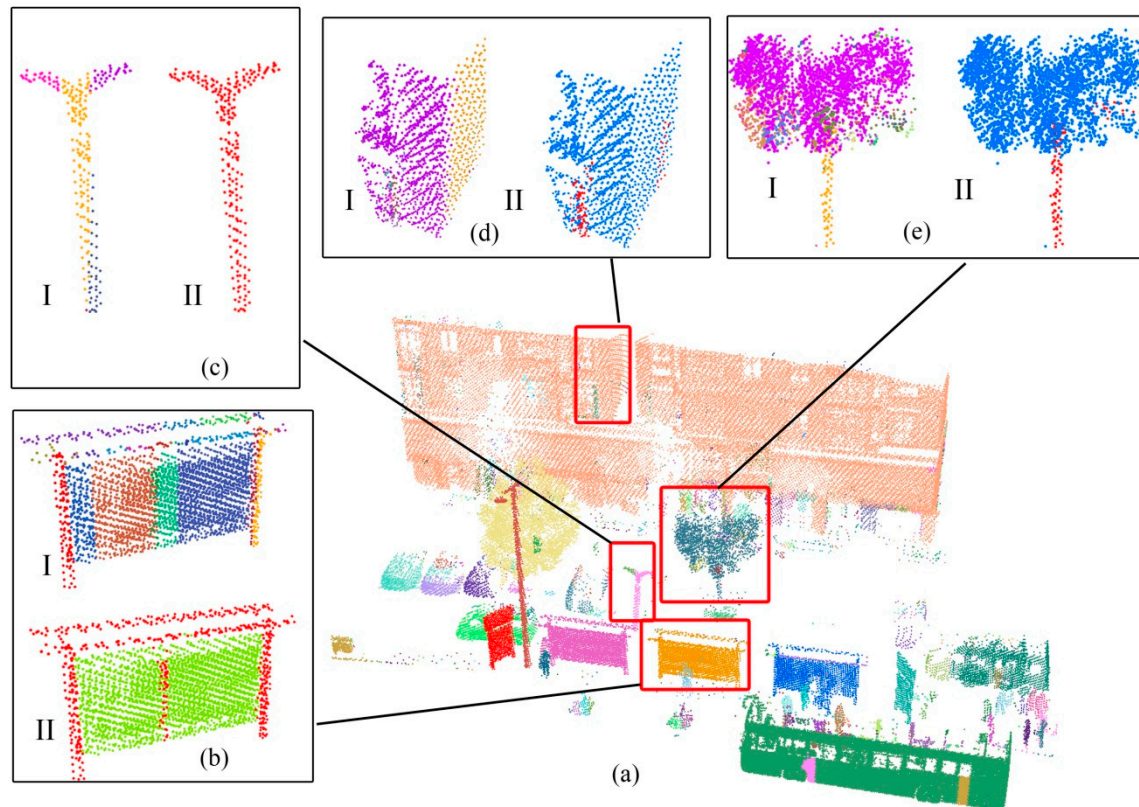
**Table 1.** Rule of merging adjacent voxel groups.

	Linear	Planar	Spherical
<b>Linear</b>	<b>If:</b> $\arccos\theta < \vec{p}_s, \vec{p}_c > \leq 10^\circ$ $\&\&  et_s - et_p  \leq T_e$ $\&\&   o_s(x, y, z) - o_p(x, y, z)   < T_o$ <b>Else if:</b> $\arccos\theta < \vec{p}_s, \vec{p}_c > \geq 80^\circ$ $\&\& S_{Min} \leq T_{md}$	<b>If:</b> $\arccos\theta < \vec{p}_s, \vec{n}_c > \leq 10^\circ$ $   \arccos\theta < \vec{p}_s, \vec{n}_c > \geq 80^\circ$ $\&\& S_{Min} \leq T_{md}$	<b>If:</b> $  o_s(x, y) - o_p(x, y)   < T_o$ $\&\& S_{Min} \leq T_{md}$
<b>Planar</b>	<b>If:</b> $\arccos\theta < \vec{n}_s, \vec{p}_c > \leq 10^\circ$ $   \arccos\theta < \vec{n}_s, \vec{p}_c > \geq 80^\circ$ $\&\& S_{Min} \leq T_{md}$	<b>If:</b> $\arccos\theta < \vec{n}_s, \vec{n}_c > \leq 10^\circ$ $\&\&  et_s - et_p  \leq T_e$ <b>Else if:</b> $\arccos\theta < \vec{n}_s, \vec{n}_c > \geq 80^\circ$ $\&\& S_{Min} \leq T_{md}$	
<b>Spherical</b>	<b>If:</b> $  o_s(x, y) - o_p(x, y)   < T_o$		<b>If:</b> $  o_s(x, y, z) - o_p(x, y, z)   < T_o$

$\vec{p}_s, \vec{n}_s, et_s$  and  $o_s(x, y, z)$  respectively denote the principal direction, surface normal, top elevation, center coordinates, and the radius of the seed voxel group. The bottom-left voxel group is usually selected as the seed voxel group. The  $\vec{p}_c, \vec{n}_c, et_c, o_c(x, y, z)$  denote the principal direction, surface normal, the top elevation and the center coordinates of the candidate voxel group in seed voxel group's neighborhood.  $S_{Min}$  represents the minimum euclidean distance between the seed voxel group and candidate voxel group, which indicate whether two candidate voxel group are connect.  $T_e, T_o$  and  $T_{md}$  are the corresponding threshold values of each condition.

The merging result of testing area without ground points as shown in Figure 6. As can be seen from Figure 6, voxel groups of one single real-world object with same or different shape can merge

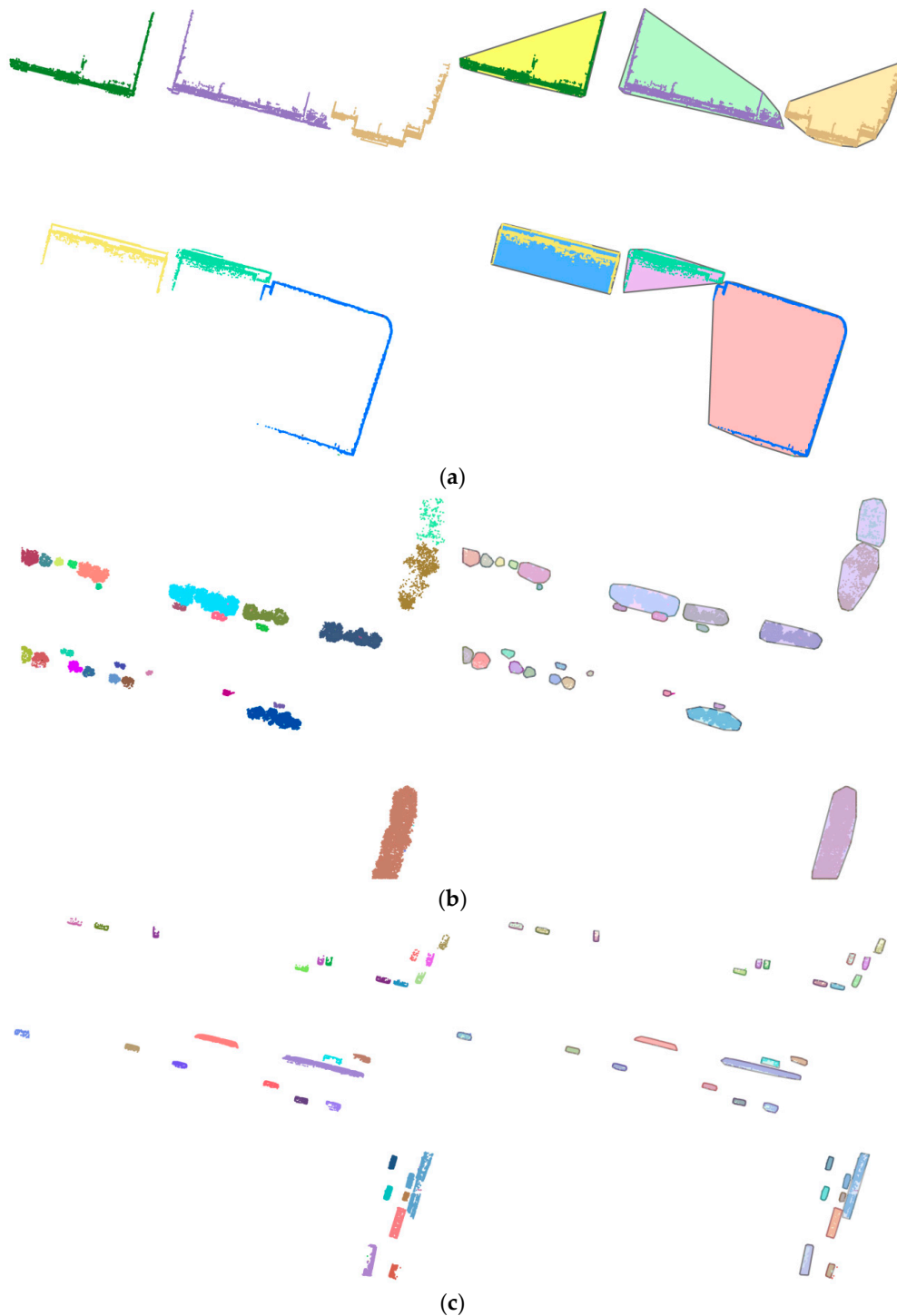
together. For example, Figure 6b shows that the parallel linear voxel groups, linear voxel groups perpendicular to each other and the planar voxel groups representing of one bus stop's components can merge well. Figure 5d shows that two planar voxel groups perpendicular to each other representing of one building's corner can merge together. Figure 6e shows that one linear voxel group representing of one tree's trunk and several spherical voxel groups representing of tree's canopy can merge well also.



**Figure 6.** Category-oriented merging. (a) Merging results, points of the same color belong to the same segment; (b–e) I: Single real-world object with several voxel groups, points of the same color belong to the same voxel group; II: Shapes of one object, red denotes linear points, green denotes surface points, and blue denotes spherical points.

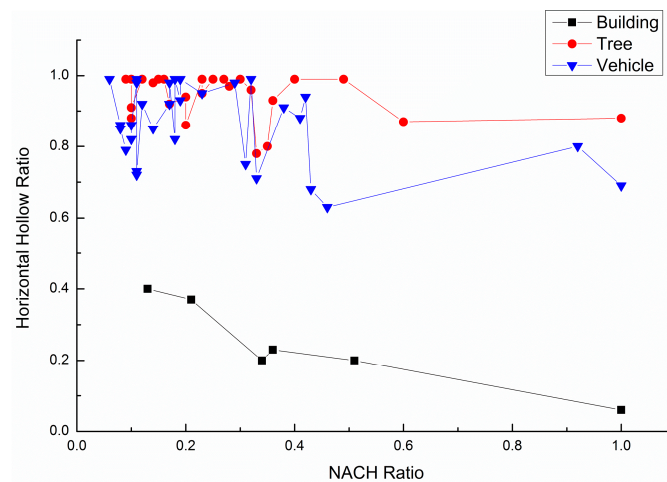
### 2.3. Horizontal Hollow Ratio-Based Building Point Identification

The laser scanner can give rich surface information of objects but couldn't penetrate the surface. Due to that fact, Aljumaily *et al.* [58] found out that each building appears as a deep hole by viewing the bottom view of the DEM generated by ALS data. In addition, he extracted building using this character. It is indicated that the defects of LiDAR data can be transformed into some advantages that can be utilized. VLS building point clouds primarily remain on the facade but are lacking in the top and inner structures because of scanner angle constraints and the resistance of building materials. Compared to objects commonly found in urban environments, such as vehicles or street trees, the ratio that the building point cloud projection area occupies, which is surrounded by the contour, is relatively small. In other words, from top view, building point clouds are more “hollow” than other object point clouds, as shown in Figure 7.



**Figure 7.** Horizontal hollow ratio-based building point identification (a–c). **Left:** top view of segments of point clouds of several buildings, trees and cars. **Right:** overlay of a convex hull and point clouds of each segment.

As shown in Figure 7, a patch with a different color represents the range of particular a convex hull generated by each segment's point clouds. It can be directly found out that the area occupied by building's point clouds is much smaller than the area of convex hull while tree and car have the opposite situation. For a more accurate analysis, the horizontal hollow ratio of every object is presented in Figure 8 to compose a graph as follows.



**Figure 8.** Horizontal hollow ratios of buildings, cars, and trees in Figure 6 (one point represents one object in Figure 6).

X-axis (NACH Ratio) represents the normalization ratio between the area of each object's convex hull and the maximum convex hull area common to this type of object. This ratio indicates the morphological changes of objects. The Y-axis represents the horizontal hollow ratio. The horizontal hollow ratio of tree is very close to the horizontal hollow ratio of car while building's horizontal hollow ratio is significantly smaller than that of the other two kinds. A detailed procedure for building points extraction based on horizontal hollow ratio can be described as follows.

**Step 1: *Extracting outline.*** The proposed method makes every segment's voxels project to the horizontal plane to form two-dimensionality grids and employ the simple and efficient method proposed by Yang [1] to extract the contour grid: if eight neighbor grids of one background grid (contains no points) are not all background grid, it will be labeled as contour grid. The aim of this step is to reduce the amount of calculation in next step.

**Step 2: *Generating convex hull.*** When get the contour grids of one segment, the convex hull of this segment is calculated by the Graham's Scan method. Furthermore, the convex hull area  $S_C$  of this segment can be calculated.

**Step 3: *Calculating horizontal hollow ratio.*** The proposed method defines the horizontal hollow ratio of each voxel cluster to indicate the above feature:

$$R_H = \frac{S_g}{S_C} \quad (12)$$

$S_g$  is the area of this segment's two-dimensionality grids:

$$S_g = N \times l \times w \quad (13)$$

where  $N$  is the number of two-dimensionality grids.

**Step 4: *Calculating threshold.*** OTSU is an automatic and unsupervised threshold selection method. Based on this method, the optimal threshold selection should be made with the best separation of the two types obtained by the threshold segmentation. The interclass separability criterion is the best statistical difference between class characteristics of maximum or minimum differences within class characteristics [59]. The building's hollow ratio is far smaller than that of other objects, as indicated by Figure 7; therefore, using the OTSU method to obtain threshold  $T$  of the hollow ratio of the divided building and other objects should achieve a good effect:

$$T = \underset{0 \leq t \leq L-1}{\text{Arg Max}} \left[ P_A(\omega_A - \omega_0)^2 + P_B(\omega_B - \omega_0)^2 \right] \quad (14)$$



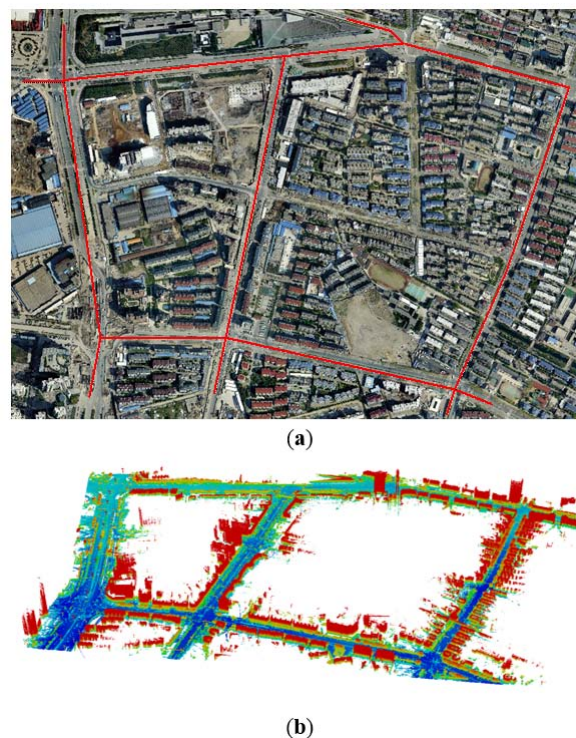
where  $L$  is the maximum value of the hollow ratio among all the voxel clusters, and  $P_A$  and  $P_B$  are the probabilities of respective building voxel clusters and non-building voxel clusters.  $\omega_A$  and  $\omega_B$  are the average hollow ratio values of building voxel clusters and non-building voxel clusters. Only voxel clusters with the average height of  $\bar{H} \geq 2.5m$  and cross-sectional area of  $Csa > 3m^2$  are used by the OTSU algorithm. If a voxel cluster's hollow ratio is less than threshold  $T$ , it is deemed a building voxel cluster and all points within it are considered building points.

### 3. Results and Discussion

The algorithm proposed was programmed in C# on Microsoft Visual Studio platform. The hardware system was a computer with 8 GB of RAM and a quad-core 2.40 GHz processor.

#### 3.1. Study Area and Experimental Data

The region of the Olympic Sports Center, Jianye District, Nanjing City, China, is chosen as the experimental area (Figure 9). To capture the vehicle LiDAR data, we employed an SSW mobile mapping system, developed by Chinese Academy of Surveying and Mapping, with a  $360^\circ$  scanning scope, a surveying range of 3 to 300 m, reflectance of 80%, and a point frequency of 200,000 points/s. The survey was conducted in 2011 and a topographic map of 1:500 was used for data correction. The overall data were approximately 4 GB, which covered a  $1.4 \text{ km} \times 0.8 \text{ km}$  area with 147,478,200 points. The point density is 270 points/ $m^2$ . Due to the amount of testing data being huge, the proposed method is unable to process it in one time. Therefore, we clip the raw testing data into 12 parts according to the road segments in practice. The experimental region contained both downtown area and urban residential area with a number of commercial and residential architecture. A shopping mall, a skyscraper, an apartment building, and a high-rise office building are the main architecture buildings in the study area. Due to the good road greening, a large number of street trees exist in the study area, which cause strong variation of point densities of building façade. On the other hand, it is sometimes difficult separate the buildings and the trees surround it.



**Figure 9.** Experimental area. (a) Aerial orthophotos of the experimental area, red line denotes the SSW mobile mapping system's driving route; (b) Raw VLS data of the experimental area.



The proposed method consists of four parts: Voxel group generating, shape recognition, category-oriented merging, and building points identification. As a few parameters and thresholds are set in each part, the summarization on the setting of the key parameters and threshold is given in Table 2. The setting basis of these thresholds and parameters includes three types: data source, calculation, and empirical. The term “data source” indicates that the values are adjusted according to the experimental data and area. The term “calculation” means that the values are calculated automatically. As for the term “empirical”, it means that the values are set empirically. In the proposed method, they are always controlled by the voxel size. The value set according to data source can be determined based on the shape property and the distribution of the objects in the experimental area. However, the optimal value set according to empirical must to be determined by several repeated test. This is the main difference among “data source” and “empirical”.

**Table 2.** The key thresholds and parameters of the proposed approach.

	Items	Values	Description	Setting Basis
Voxel group generating	$S$	0.5 m	The voxel size	Empirical
	$T_S$	0.2 m	To divide the adjacent voxel in vertical direction	Data source
	$T_{End}$	0.85	To terminate the growth of voxel groups' generating	Chen <i>et al.</i> [11]
Shape recognition	$N_p$	5 pts	Minimum number of points for PCA	Empirical
	$R_i$	0.1 m	The increment of the search radius	Empirical
Category-oriented merging	$T_e$	0.1 m	Maximal difference of elevation between two voxel groups	Data source
	$T_o$	0.5 m	Maximal distance between two voxel groups' center	Empirical
	$T_{md}$	0.15 m	Maximal minimum euclidean distance between two voxel groups	Data source
Building point identification	$T$	Automatic	The threshold of the horizontal hollow ratio to identify building points	Calculation
	$\bar{H}$	2.5 m	Minimum average height of voxel cluster	Data source
	$Csa$	3 m <sup>2</sup>	Minimum Cross-sectional area of voxel cluster	Data source

During the generating of voxel group, we set the size  $S = 0.5$  m to form a regular 3-D voxel grid. The size of the voxel is closely related to the extraction of the building points. Some of the buildings lost part of the facade and other appendants, like the podium building, when the voxel size is too small. If the voxel size chosen is too large, the tree and brush close to the building will be classified to building points incorrectly. Some small and low-rise building will be identified as tree or brush due to the same reason. On the other hand, the smaller the voxel size is, the long the execution time is. The threshold to divide the adjacent voxel in vertical direction is set according to the data source. The threshold to terminate the growth of voxel groups' generating is set according to Chen *et al.* [11]. By this test, 0.85 is the best value.

During the shape recognition of the points in each voxel group, the minimal number of points to initialize the minimum search radius and the increment of the search radius is set empirically. The voxel size is one influential factor of the values.

During the category-oriented merging with voxel groups, voxel groups are merged by utilizing the shape information on several rules. The maximal difference in elevation, maximal minimum Euclidean distance between two voxel groups can be setting according to the data source. The maximal distance between two voxel groups' centers can be set empirically. This value is depending on the voxel size. The bigger the voxel size was set, the greater the value.

During the building point identification, the horizontal hollow ratio of each voxel cluster meets the conditions is calculated. The threshold of the horizontal hollow ratio to identify building points is calculated automatically by the OTSU method. The minimum average height and the minimum cross-sectional area of voxel clusters need to be set according to the data source. These two values are the approximate size of a newsstand, which can be often viewed along the urban street of Nanjing and be considered as the smallest building.

### 3.2. Extraction Results of Building Points

The raw vehicle LiDAR point clouds and aerial orthophotos are used to manually extract the building points. The aerial orthophotos were acquired at the same time period when collecting LiDAR data and they can be matched. The spatial resolution of the aerial orthophotos is 0.2 m and presented in 3 bands red-green-blue (RGB). The extraction results were then used as the validation data. Building point boundaries were connected to determine the building regions as the ground truth, referred to herein as the true building region. Figure 10 illustrates the extraction results of building points. LiDAR points are marked in blue for buildings, red for the ground, and green for the other features. As shown in Figure 10b–d, our method extracted not only high-rise and low-rise buildings, but also buildings with special exteriors or complex structures. Moreover, as shown in Figure 10e, the proposed method recognized buildings with missing parts of facade that were caused by field measurement conditions or equipment factors. The SSW vehicle-borne laser scanning system accomplished the objective of obtaining 192 relatively complete buildings from the experimental area with a total of 235 buildings. A relatively complete building refers to the scanning point cloud of the building with at least two facades. The types of buildings in the study area included low cottages, middle-rise residential buildings, high-rise commercial buildings, and unique structures, such as bell towers and theaters.

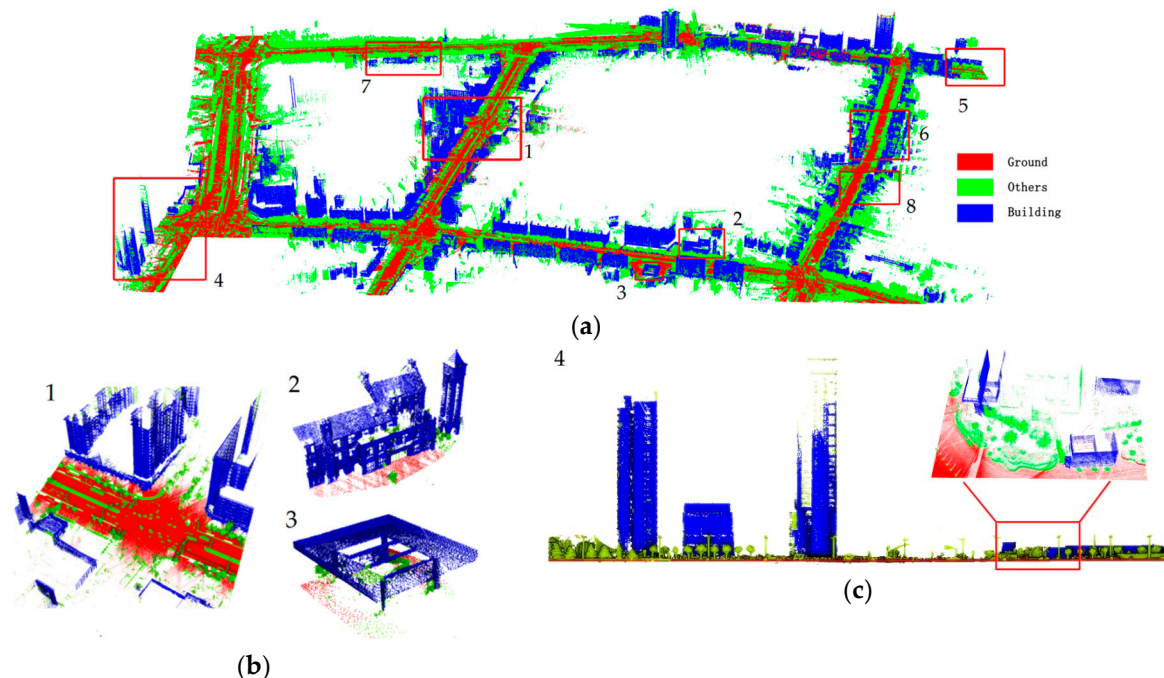
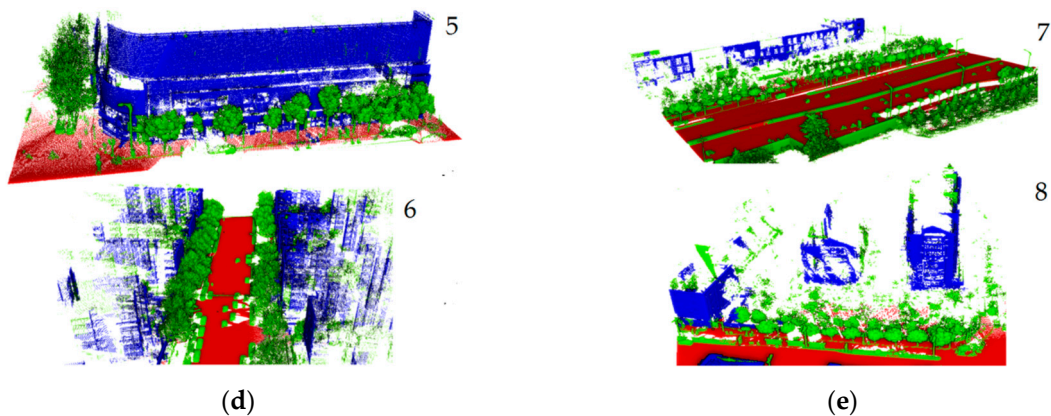


Figure 10. Cont.



**Figure 10.** Building point extraction results. (a) Extraction results of buildings in the experiment region; (b,c) Proposed method successfully detected various building shapes, including skyscrapers and low cottages; (d) Proposed method effectively separated a building and the trees attached to it; (e) Results show that the method could also recognize buildings with sparse LiDAR points or lack of partial structures.

### 3.3. Evaluation of Extraction Accuracy

In this section, the extraction accuracy of the proposed method is validated from two aspects: building-based evaluation for overall experimental area (a whole building taken as an evaluation unit) and point-based evaluation for individual building (a point taken as an evaluation unit). To analyze the proposed algorithm in details, this article divided the buildings into three types: low- (one to two stories), middle- (three to seven stories), and high-rise structures (more than seven stories).

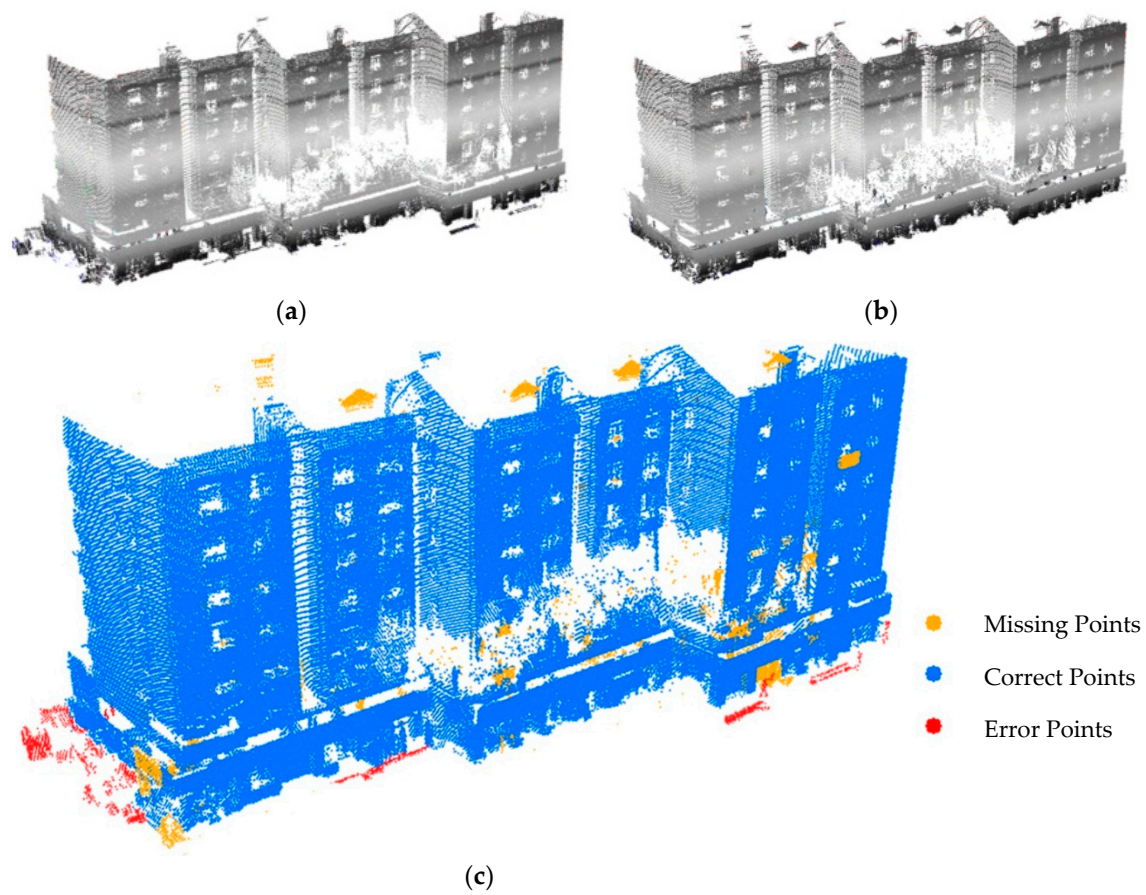
The correctness and completeness of the method are used as indexes for the evaluation, as follows:

$$Completeness = \frac{TP}{TP + FN} \quad (15)$$

$$Correctness = \frac{TP}{TP + FP} \quad (16)$$

In the evaluation for overall experimental area, where TP is the number of true buildings, FP is the number of wrong buildings, and FN is the number of mis-detected buildings. The buildings derived by manual operation are taken as reference data. By overlaying one extracted building and the corresponding reference data, the overlapped area of them is calculated. If the ratio of this overlapped area to area of the extracted building is larger than 70%, the extracted building is taken as true one. Otherwise, it is considered as a wrong one. If a building is not be detected by the automatic process, it is considered to be a mis-detected building.

In the evaluation for individual building, where TP is the number of true points belonging to it, FP is the number of wrong points belonging to it, and FN is the number of mis-detected points belonging to it. The extracted building points and true building points are put together. The points of overlapped region are regarded as true points (blue points in Figure 11c); points existing only in the truth regions are deemed misdetections (yellow points in Figure 11c), and points existing only in the extracted regions are considered wrong points (red points in Figure 11c).



**Figure 11.** Point-based evaluation for individual building. (a) Automatic extraction results of a building; (b) Manual extraction results of the same building; (c) Overlay result with the correct, error, and missing points denoted in blue, red, and yellow, respectively.

### 3.3.1. Building-Based Evaluation for Overall Experimental Area

There are 192 buildings involved in the evaluation in this section, including 42 low-rise buildings, 126 medium-rise buildings, and 24 high-rise buildings.

As the height of the building increased, the completeness of extraction results increased; the high-rise buildings' completeness reached 100%, whereas that of the low-rise buildings reached only 86.3%. This was due to the low-rise buildings' top and internal structures being more apt to scanning by the mobile laser scanner; therefore, the low-rise buildings' hollow ratios calculated by the algorithm reached lower values than those of the high-rise buildings. Hence, if a very tall building and a low-rise building were in the same region, the low-rise building with a low hollow ratio may have been incorrectly identified as a vehicle, bush, or other object. The overall completeness and correctness were 94.7% and 91%, respectively. As a comparison, one automatic building detection method reported by Truong-Hong and Laefer [41] conducted in the similarly dense urban area achieved 95.1% completeness and 67.7% correctness from ALS data.

Two primary types of incorrect detections occurred. Firstly, because some vehicles were in high-speed motion, these vehicles' point clouds were stretched, or adjacent vehicles' point clouds may have merged, which caused them to be incorrectly detected as buildings owing to the high hollow ratio values they incurred. Secondly, because their shapes can be similar to building facades, high fences resulted in high hollow ratios and they were consequently incorrectly detected as buildings.



### 3.3.2. Point-Based Evaluation for Individual Building

In this evaluation, to further evaluate the effect of the proposed method, complex buildings are entered into a separate category (a new type of buildings), named complex buildings in Table 2. The individual evaluation results in Table 3 show that the middle-rise buildings have better extraction results than the low-rise and high-rise buildings. The average completeness and correctness of the middle-rise buildings are 95% and 95.7%, respectively. The low-rise buildings yield the worst extraction results, with a completeness and correctness of 94.8% and 93.1%, respectively. These latter results are due to the relative complexity of the vicinities of low-rise buildings; the proposed method employs the voxel group to segment the point clouds. This combined some of the buildings' point clouds and those of adjacent bushes, trees, and other objects into one group, which reduced the accuracy.

**Table 3.** Completeness and correctness of the extracted building points.

Type	Number of Points			Completeness (%)	Correctness (%)	Average Com (%)	Average Corr (%)
	TP	FN	FP				
Low-rise	15,744	500	1239	96.9	92.7	94.8	93.1
	54,399	2233	349	96.1	99.4		
	6750	0	598	100	91.9		
	6830	377	135	94.8	98.1		
	30,752	3234	3827	90.5	88.9		
	38,580	0	5122	100	88.3		
	20,751	1705	512	92.4	97.6		
	8048	0	1147	100	87.5		
	23,606	3234	336	87.7	98.6		
	12,083	1473	1639	89.1	88.1		
Medium-rise	167,478	934	2126	99.4	98.7	95.0	95.7
	85,670	543	1408	99.4	98.4		
	194,255	1560	3210	99.2	98.4		
	198,123	846	1042	99.6	99.5		
	125,507	6835	773	94.8	99.4		
	237,798	11,732	10,592	95.3	95.7		
	50,687	10,466	5872	82.9	89.6		
	219,639	9897	5396	95.7	97.6		
	45,340	3699	1146	92.5	97.5		
	25,536	2229	5587	92.0	82.0		
High-rise	115,343	14,306	388	89.0	99.7	91.0	99.4
	186,558	6697	2993	96.5	98.4		
	253,489	14,368	1152	94.6	99.5		
	206,176	6467	1388	97.0	99.3		
	209,904	38,477	3387	84.5	98.4		
	320,217	26,779	432	92.3	99.9		
	153,428	26,186	0	85.4	100.0		
	144,498	10,957	0	93.0	100.0		
	54,596	9874	0	84.7	100.0		
	133,353	9248	652	93.5	99.5		
Complex	313,922	22,428	1716	93.3	99.5	91.9	99.0
	34,455	1798	254	95.0	99.3		
	26,540	739	613	97.3	97.7		
	11,945	4250	0	73.8	100.0		
	17,711	2415	136	88.0	99.2		
	608,188	24,904	0	96.1	100.0		
	281,385	26,653	342	91.3	99.9		
	282,115	11,010	2341	96.2	99.2		
	19,957	832	687	96.0	96.7		
	312,765	27,144	4336	92.0	98.6		

The high-rise buildings demonstrate the highest correctness and lowest completeness at 99.4% and 91%, respectively. This is due to the limitation of the scanner angle; *i.e.*, the vehicle-borne laser



scanning system was unable to obtain the point clouds of upper stories and facades that did not face the road. This prevented some of the building point clouds being added to the main part of the building in regional growing process. On the other hand, for high-rise buildings, such as commercial and office buildings, the surrounding environments were relatively simple; objects adjacent to these buildings had very distinct elevations and shapes. Therefore, the high-rise building extraction results had the lowest completeness but highest correctness.

The complex buildings have an irregular shape and complicated structure, so it is hard to identify all the parts of a complex building. The complex building's average completeness is 91.9%, on the whole in a satisfactory level. However, from the ten chosen complex building, it is can be found find that the completeness of each building has a large span, from 73.8% to 97.3%, which indicate that the method may face a challenge when dealing with some special building.

### 3.4. Experiment Discussion

Overall, the detection results in Figure 10 and the evaluation outcomes showed that the proposed method can work well in dense urban areas. The voxels generated by the raw point clouds were divided into voxel columns. A full  $\lambda$ -Schedule algorithm, which considers the complexity of the object boundary, was used for the voxel columns merging and an optimal regularization parameter can be drawn. As a contrast, the termination rules of a 3D region growing approach are difficult to determine. Although the 3D region growing method can construct voxel groups more efficient but the results are always over-segmentation. Five combinations based on two voxel groups' shapes were presented. Each combination has its own merging rule and high-precise segmentation results were achieved by region growing. The method have more potential for acquiring a further improvement by using machine learning (e.g., Support Vector Machine, SVM) and fusing with intensity or color information of point clouds.

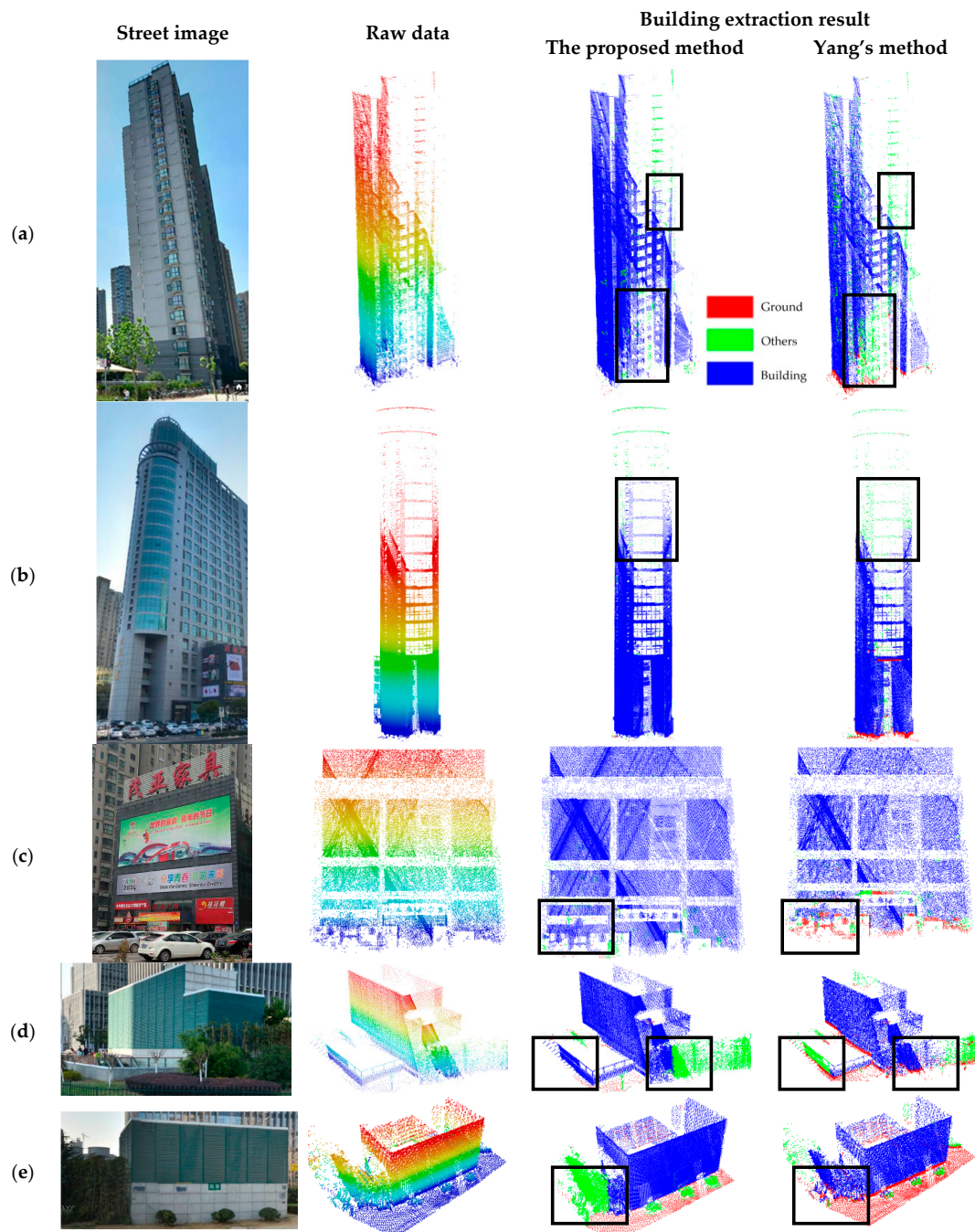
The proposed method and the method of Yang *et al.* [37] show some similarity in the strategy of the segmentation of point clouds, so a comparison was undertaken in this section. The method of Yang *et al.* [37] generates multi-scale supervoxels from VLS data and segments these supervoxels. Then a set rules was defined for merging adjacent segments with semantic knowledge. We divided both the strategy of the segmentation of point clouds of the proposed method and Yang's method into three steps: the point organization, shape recognition and merging. Region 5 (red rectangle in Figure 10a) with 458,259 points is the test area for the comparison between the proposed method and Yang's method. Point clouds are assigned to 1400 voxel groups or 7282 supervoxels. It can be found that the objects with simple structure, like building facade, have greatly fewer voxel groups than supervoxels. There are many more small areas of flat facade of the building were classified into linear or spherical structure incorrectly by Yang's method. This is due to that the voxel group is bigger than supervoxel in size, so the voxel group can contain more complete structure of the single real-world object. The points of the building are divided into one segment by the proposed method while they were separated into several segments by Yang's method. Through the comparison of the two methods' result, the proposed method handles the shape recognition and segmentation of building facade better than the method of Yang *et al.* [37]. Table 4 shows that the proposed method needs less execution time than Yang's method in each step.

**Table 4.** Time performance of the proposed method and the method of Yang *et al.* [37].

	Point Organization	Shape Recognition	Merging	Total
The proposed method(s)	4.32	9.91	9.45	23.68
Yang's method(s)	7.67	10.44	16.96	35.07

Figure 12 illustrates the comparative experiments result of building extraction by the proposed method and the method of Yang *et al.* [37]. Region 4 (red rectangle in Figure 9a) is the test area with

5,298,460 points and five buildings. Both methods extract all buildings and no misclassification error has occurred. Figure 12a–c shows that the proposed method can extract more details of the building (see black rectangles), like beam and bay window. Figure 12d,e shows that the proposed method can reduce the error caused by the noise around the building (e.g., bush, tree, and car) and as far as possible to retain the accessory structures of the building at the same time. On the other hand, Yang’s method has wrongly identified the bottom of all the extracted buildings as ground points while the proposed method avoids this mistake.



**Figure 12.** Comparison of building extraction result between the proposed method and the method of Yang *et al.* [37]. (a–e) Left to right: street image, raw VLS data, the result by the proposed method and the result by Yang’s method of the specific building.

#### 4. Conclusions

The study proposed an approach for automatic extraction of building points using vehicle-borne laser scanning data. From experimental results, the following points can be drawn.

(1) The proposed approach, including voxel group-based shape recognition, category-oriented merging, and horizontal hollow-based building point identification, can be applied in the various types of buildings, in large-scale and complex urban environments.

(2) The category-oriented merging with voxel group-based shape recognition is effective in improving the accuracy of segmentation and to develop the new voxel group structure for accelerating processing and simplifying shape recognition.

(3) The concept of horizontal hollow ratio for building point cloud identification can accurately extract various forms of buildings, from low cottages and common residential buildings to the towering skyscrapers and remarkable stylish theaters, without requiring complex semantic rules. This point indicated that some characteristics of LiDAR data caused by the constraints of the laser sensor are not obstacles, but some good indicators for particular objects. This is a new concept in the classification of LiDAR data fields.

Our future work will focus on fusing VLS data with ALS data and optical images, such as street images, to obtain more complete and accurate building point detection results. Otherwise, the huge data amount of mobile LiDAR data largely reduces the efficiency of information extraction for specific applications. Future work is to develop high-performance computation technology for mobile LiDAR data processing.

**Acknowledgments:** This work is supported by the National Natural Science Foundation of China (Grant No. 41371017, 41001238), the National Key Technology R&D Program of China (Grant No. 2012BAH28B02). The authors thank the anonymous reviewers and members of the editorial team for their comments and contributions.

**Author Contributions:** Yu Wang and Liang Cheng proposed the core idea of voxel group and horizontal hollow analysis based building points extraction method and designed the technical framework of the paper. Yu Wang performed the data analysis, results interpretation, and manuscript writing. Yang Wu assisted with the experimental result analysis. Yanming Chen and Manchun Li assisted with refining the research design and manuscript writing.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	linear dichroism

#### References

1. Yang, B.; Wei, Z.; Li, Q.; Li, J. Semiautomated building facade footprint extraction from mobile LiDAR point clouds. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 766–770. [[CrossRef](#)]
2. Hong, S.; Jung, J.; Kim, S.; Cho, H.; Lee, J.; Heo, J. Semi-automated approach to indoor mapping for 3D as-built building information modeling. *Comput. Environ. Urban Syst.* **2015**, *51*, 34–46. [[CrossRef](#)]
3. Remondino, F. Heritage recording and 3D modeling with photogrammetry and 3D scanning. *Remote Sens.* **2011**, *3*, 1104–1138. [[CrossRef](#)]
4. Cheng, L.; Wang, Y.; Chen, Y.; Li, M. Using LiDAR for digital documentation of ancient city walls. *J. Cult. Herit.* **2016**, *17*, 188–193. [[CrossRef](#)]
5. He, M.; Zhu, Q.; Du, Z.; Hu, H.; Ding, Y.; Chen, M. A 3D shape descriptor based on contour clusters for damaged roof detection using airborne LiDAR point clouds. *Remote Sens.* **2016**, *8*, 189. [[CrossRef](#)]
6. Cheng, L.; Gong, J.; Li, M.; Liu, Y. 3D building model reconstruction from multi-view aerial imagery and LiDAR data. *Photogramm. Eng. Remote Sens.* **2011**, *77*, 125–139. [[CrossRef](#)]

7. Wu, B.; Yu, B.; Yue, W.; Shu, S.; Tan, W.; Hu, C.; Huang, Y.; Wu, J.; Liu, H. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* **2013**, *5*, 584–611. [[CrossRef](#)]
8. Rutzinger, M.; Rottensteiner, F.; Pfeifer, N. A comparison of evaluation techniques for building extraction from airborne laser scanning. *IEEE J. Sel. Topics Appl. Earth Observ. Remote.* **2009**, *2*, 11–20. [[CrossRef](#)]
9. Sampath, A.; Shan, J. Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1554–1567. [[CrossRef](#)]
10. Kim, K.; Shan, J. Building roof modeling from airborne laser scanning data based on level set approach. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 484–497. [[CrossRef](#)]
11. Yanming, C.; Liang, C.; Manchun, L.; Jiechen, W.; Lihua, T.; Kang, Y. Multiscale grid method for detection and reconstruction of building roofs from airborne LiDAR data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2014**, *7*, 4081–4094.
12. Sun, S.; Salvaggio, C. Aerial 3D building detection and modeling from airborne LiDAR point clouds. *IEEE J. Sel. Topics Appl. Earth Observ. Remote.* **2013**, *56*, 1440–1449. [[CrossRef](#)]
13. Rottensteiner, F. Automatic generation of high-quality building models from LiDAR data. *IEEE Comput. Graphics Appl.* **2003**, *23*, 42–50. [[CrossRef](#)]
14. Li, Q.; Chen, Z.; Hu, Q. A model-driven approach for 3D modeling of pylon from airborne LiDAR data. *Remote Sens.* **2015**, *7*, 11501–11524. [[CrossRef](#)]
15. Wang, R. 3D building modeling using images and LiDAR: A review. *Int. J. Image Data Fusion* **2013**, *4*, 273–292. [[CrossRef](#)]
16. Xu, H.; Cheng, L.; Li, M.; Chen, Y.; Zhong, L. Using octrees to detect changes to buildings and trees in the urban environment from airborne LiDAR data. *Remote Sens.* **2015**, *7*, 9682–9704. [[CrossRef](#)]
17. Xu, S.; Vosselman, G.; Oude Elberink, S. Detection and classification of changes in buildings from airborne laser scanning data. *Remote Sens.* **2015**, *7*, 17051–17076. [[CrossRef](#)]
18. Pu, S.; Rutzinger, M.; Vosselman, G.; Oude Elberink, S. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S28–S39. [[CrossRef](#)]
19. Vo, A.V.; Truong-Hong, L.; Laefer, D.F. Aerial laser scanning and imagery data fusion for road detection in city scale. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015.
20. Tong, L.; Cheng, L.; Li, M.; Wang, J.; Du, P. Integration of LiDAR data and orthophoto for automatic extraction of parking lot structure. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2014**, *7*, 503–514. [[CrossRef](#)]
21. Cheng, L.; Zhao, W.; Han, P.; Zhang, W.; Shan, J.; Liu, Y.; Li, M. Building region derivation from LiDAR data using a reversed iterative mathematic morphological algorithm. *Opt. Commun.* **2013**, *286*, 244–250. [[CrossRef](#)]
22. Mongus, D.; Lukač, N.; Žalik, B. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 145–156. [[CrossRef](#)]
23. Chun, L.; Beiqi, S.; Xuan, Y.; Nan, L.; Hangbin, W. Automatic buildings extraction from LiDAR data in urban area by neural oscillator network of visual cortex. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2013**, *6*, 2008–2019.
24. Berger, C.; Voltersen, M.; Hese, S.; Walde, I.; Schmulilius, C. Robust extraction of urban land cover information from HSR multi-spectral and LiDAR data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2013**, *6*, 2196–2211. [[CrossRef](#)]
25. Zhang, J.; Lin, X. Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing. *Int. J. Image Data Fusion* **2016**, *2016*, 1–31. [[CrossRef](#)]
26. Becker, S.; Haala, N. Grammar supported facade reconstruction from mobile LiDAR mapping. *Int. Arch. Photogramm. Remote Sens.* **2009**, *38*, 13–16.
27. Chan, T.O.; Lichti, D.D.; Glennie, C.L. Multi-feature based boresight self-calibration of a terrestrial mobile mapping system. *ISPRS J. Photogramm. Remote Sens.* **2013**, *82*, 112–124. [[CrossRef](#)]
28. Manandhar, D.; Shibasaki, R. Auto-extraction of urban features from vehicle-borne laser data. *Int. Arch. Photogramm. Remote Sens.* **2002**, *34*, 650–655.



29. Hammoudi, K.; Dornaika, F.; Soheilian, B.; Paparoditis, N. Extracting outlined planar clusters of street facades from 3D point clouds. In Proceedings of the 2010 Canadian Conference on Computer and Robot Vision (CRV), Ottawa, ON, Canada, 31 May–2 June 2010.
30. Jochem, A.; Höfle, B.; Rutzinger, M. Extraction of vertical walls from mobile laser scanning data for solar potential assessment. *Remote Sens.* **2011**, *3*, 650–667. [[CrossRef](#)]
31. Tarsha-Kurdi, F.; Landes, T.; Grussenmeyer, P. Hough-transform and extended ransac algorithms for automatic detection of 3D building roof planes from LiDAR data. *Proc. ISPRS* **2007**, *36*, 407–412.
32. Rutzinger, M.; Elberink, S.O.; Pu, S.; Vosselman, G. Automatic extraction of vertical walls from mobile and airborne laser scanning data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2009**, *38*, W8.
33. Moosmann, F.; Pink, O.; Stiller, C. Segmentation of 3D LiDAR data in non-flat urban environments using a local convexity criterion. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009.
34. Munoz, D.; Vandapel, N.; Hebert, M. Directional associative markov network for 3D point cloud classification. In Proceedings of the Fourth International Symposium on 3D Data Processing, Visualization and Transmission, Atlanta, GA, USA, 18 June 2008.
35. Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality based scale selection in 3D LiDAR point clouds. *Proc. ISPRS* **2011**, *38*, W52. [[CrossRef](#)]
36. Yang, B.; Dong, Z. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 19–30. [[CrossRef](#)]
37. Yang, B.; Dong, Z.; Zhao, G.; Dai, W. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2015**, *99*, 45–57. [[CrossRef](#)]
38. Li, B.; Li, Q.; Shi, W.; Wu, F. Feature extraction and modeling of urban building from vehicle-borne laser scanning data. *Proc. ISPRS* **2004**, *35*, 934–939.
39. Aijazi, A.K.; Checchin, P.; Trassoudaine, L. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. *Remote Sens.* **2013**, *5*, 1624–1650. [[CrossRef](#)]
40. Yang, B.; Wei, Z.; Li, Q.; Li, J. Automated extraction of street-scene objects from mobile LiDAR point clouds. *Int. J. Remote Sens.* **2012**, *33*, 5839–5861. [[CrossRef](#)]
41. Truong-Hong, L.; Laefer, D.F. Quantitative evaluation strategies for urban 3D model generation from remote sensing data. *Comput. Graphics* **2015**, *49*, 82–91. [[CrossRef](#)]
42. Weishampel, J.F.; Blair, J.; Knox, R.; Dubayah, R.; Clark, D. Volumetric LiDAR return patterns from an old-growth tropical rainforest canopy. *Int. J. Remote Sens.* **2000**, *21*, 409–415. [[CrossRef](#)]
43. Riano, D.; Meier, E.; Allgöwer, B.; Chuvieco, E.; Ustin, S.L. Modeling airborne laser scanning data for the spatial generation of critical forest parameters in fire behavior modeling. *Remote Sens. Environ.* **2003**, *86*, 177–186. [[CrossRef](#)]
44. Chasmer, L.; Hopkinson, C.; Treitz, P. Assessing the three-dimensional frequency distribution of airborne and ground-based LiDAR data for red pine and mixed deciduous forest plots. *Proc. ISPRS* **2004**, *36*, 8W.
45. Popescu, S.C.; Zhao, K. A voxel-based LiDAR method for estimating crown base height for deciduous and pine trees. *Remote Sens. Environ.* **2008**, *112*, 767–781. [[CrossRef](#)]
46. Reitberger, J.; Schnörr, C.; Krzystek, P.; Stilla, U. 3D segmentation of single trees exploiting full waveform LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 561–574. [[CrossRef](#)]
47. Wang, C.; Tseng, Y. Dem generation from airborne LiDAR data by an adaptive dual-directional slope filter. *Proc. ISPRS* **2010**, *38*, 628–632.
48. Jwa, Y.; Sohn, G.; Kim, H. Automatic 3D powerline reconstruction using airborne LiDAR data. *Int. Arch. Photogramm. Remote Sens.* **2009**, *38*, 105–110.
49. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LiDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011.
50. Park, H.; Lim, S.; Trinder, J.; Turner, R. Voxel-based volume modelling of individual trees using terrestrial laser scanners. In Proceedings of the 15th Australasian Remote Sensing and Photogrammetry Conference, Alice Springs, Australia, 13–17 September 2010.
51. Hosoi, F.; Nakai, Y.; Omasa, K. 3D voxel-based solid modeling of a broad-leaved tree for accurate volume estimation using portable scanning LiDAR. *ISPRS J. Photogramm. Remote Sens.* **2013**, *82*, 41–48. [[CrossRef](#)]



52. Stoker, J. Visualization of multiple-return LiDAR data: Using voxels. *Photogramm. Eng. Remote Sens.* **2009**, *75*, 109–112.
53. Liang, C.; Yang, W.; Yu, W.; Lishan, Z.; Yanming, C.; Manchun, L. Three-dimensional reconstruction of large multilayer interchange bridge using airborne LiDAR data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2015**, *8*, 691–708.
54. Cheng, L.; Tong, L.; Wang, Y.; Li, M. Extraction of urban power lines from vehicle-borne LiDAR data. *Remote Sens.* **2014**, *6*, 3302–3320. [[CrossRef](#)]
55. Redding, N.J.; Crisp, D.J.; Tang, D.; Newsam, G.N. An efficient algorithm for mumford-shah segmentation and its application to sar imagery. In *Proceedings of the 1999 Conference on Digital Image Computing: Techniques and Applications*, Perth, Australia; 1999; pp. 35–41.
56. Robinson, D.J.; Redding, N.J.; Crisp, D.J. *Implementation Of A Fast Algorithm For Segmenting Sar Imagery*; DSTO-TR-1242; Defense Science and Technology Organization: Sydney, Australia, 2002.
57. Korah, T.; Medasani, S.; Owechko, Y. Strip histogram grid for efficient LiDAR segmentation from urban environments. In *Proceedings of the 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Colorado Springs, CO, USA, 20–25 June 2011.
58. Aljumaily, H.; Laefer, D.F.; Cuadra, D. Big-data approach for 3D building extraction from aerial laser scanning. *J. Comput. Civil Eng.* **2015**, *30*, 04015049. [[CrossRef](#)]
59. Otsu, N. A threshold selection method from gray-level histograms. *Automatica* **1975**, *11*, 23–27.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).