



## Article

# Extracting Urban Road Footprints from Airborne LiDAR Point Clouds with PointNet++ and Two-Step Post-Processing

Haichi Ma <sup>1</sup>, Hongchao Ma <sup>1,2,\*</sup>, Liang Zhang <sup>3</sup>, Ke Liu <sup>1</sup> and Wenjun Luo <sup>1</sup>

<sup>1</sup> School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; mahaichi@whu.edu.cn (H.M.); Liu\_ke@whu.edu.cn (K.L.); yunitongzai@whu.edu.cn (W.L.)

<sup>2</sup> Department of Oceanography, Dalhousie University, Halifax, NS B3H 4R2, Canada

<sup>3</sup> Faculty of Resources and Environmental Science, Hubei University, Wuhan 430062, China; zhangliang@hubu.edu.cn

\* Correspondence: hchma@whu.edu.cn; Tel.: +86-138-0864-8104

**Abstract:** In this paper, a novel framework for the automatic extraction of road footprints from airborne LiDAR point clouds in urban areas is proposed. The extraction process consisted of three phases: The first phase is to extract road points by using the deep learning model PointNet++, where the features of the input data include not only those selected from raw LiDAR points, such as 3D coordinate values, intensity, etc., but also the digital number (DN) of co-registered images and generated geometric features to describe a strip-like road. Then, the road points from PointNet++ were post-processed based on graph-cut and constrained triangulation irregular networks, where both the commission and omission errors were greatly reduced. Finally, collinearity and width similarity were proposed to estimate the connection probability of road segments, thereby improving the connectivity and completeness of the road network represented by centerlines. Experiments conducted on the Vaihingen data show that the proposed framework outperformed others in terms of completeness and correctness; in addition, some narrower residential streets with 2 m width, which have normally been neglected by previous studies, were extracted. The completeness and the correctness of the extracted road points were 84.7% and 79.7%, respectively, while the completeness and the correctness of the extracted centerlines were 97.0% and 86.3%, respectively.

**Keywords:** airborne LiDAR; road footprints; geometric features; road centerline



**Citation:** Ma, H.; Ma, H.; Zhang, L.; Liu, K.; Luo, W. Extracting Urban Road Footprints from Airborne LiDAR Point Clouds with PointNet++ and Two-Step Post-Processing. *Remote Sens.* **2022**, *14*, 789. <https://doi.org/10.3390/rs14030789>

Academic Editor: Saeid Himayouni

Received: 22 December 2021

Accepted: 6 February 2022

Published: 8 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Airborne light detection and ranging (LiDAR) is an active remote sensing technique for acquiring 3D geospatial data over the Earth's surface [1,2], with which a point clouds dataset encoding 3D coordinate values under a given geographic coordinate system can be generated [3]. The point clouds can be further processed to extract thematic information and to generate geo-mapping products, such as man-made objects [4], stand-alone plants [5], a digital elevation model (DEM) [6], etc. The latest commercial LiDAR systems, such as the Leica ALS80 airborne LiDAR, can provide a planar accuracy better than 10 cm and a vertical accuracy better than 6 cm at a typical flying height of 1000 m.

LiDAR point clouds are becoming one of the main data sources for road extraction because of at least four advantages compared with optical imagery. First, the elevation information from LiDAR points can be used to separate elevated structures, such as buildings and vegetation from roads [7,8]. Second, LiDAR points can cover an urban road continuously. The ability of LiDAR's penetration through tree canopies reduces the occlusion effect caused by trees; its relatively narrow scanning angle also ensures the integrity of road information [9]. Third, LiDAR data contains extra information, such as intensity and full-waveform, which help to improve the accuracy of road extraction [10]. Finally, as acquired by an active sensor, LiDAR data are not affected by shadow and are less likely to be affected by weather [11].

Road extraction from LiDAR data has been a research topic since the late 1990s. A variety of approaches have been proposed and explored since then. Rieger et al. [12] extracted roads in forested areas from LiDAR data, and used the extracted roads to enhance the quality of a digital terrain model (DTM). Clode et al. [13] proposed a hierarchical classification method to convert LiDAR data into a digital surface model (DSM) to extract road points from point clouds. The road candidate points were obtained by filtering from a given distance to the DTM and intensity value. Road patches were later connected to a road network through a morphological closing operation. However, the correctness of this method was not high, especially when a road was connected to a bridge. To improve the efficiency of this method, Clode et al. [14] introduced another algorithm called phase coded disk (PCD) to vectorize the road centerlines and calculate the road width. Choi et al. [15] demonstrated that the integration of height, reflectance, and geometric information of roads was a crucial factor for reliable and correct classification of road points. Samadzadegan et al. [16] proposed a road extraction method from point clouds based on combining multiple classifiers. In their proposed method, majority voting and selective naive Bayesian were used for the fusion of the results of multiple classifiers. Zhu and Mordohai [17] formulated road extraction as a minimum coverage problem, then solved it in the presence of large-scale range datasets. Zhao and You [18] designed structure templates to search for roads on ground intensity images, and road widths and orientations were determined by a subsequent voting scheme. Matkan et al. [19] classified the LiDAR data into five classes: roads, trees, buildings, grassland, and cement using support vector machines (SVM) classification, where a method based on Radon transform and Spline interpolation was employed to automatically locate and fill the gaps and holes in the road network; this showed acceptable performance under straight road conditions, but did not perform well at intersections. Li et al. [20] presented a novel algorithm for road detection from airborne LiDAR point clouds adaptively for the variability of intensity data of the road network. The candidate road points were identified from ground points by a local intensity distribution histogram, while the final road points were verified by global inference based on the roughness and area of the road candidate point sets. Hui et al. [21] proposed a method called SRH, which was composed of three key algorithms: skewness balancing, rotating neighborhood, and hierarchical fusion and optimization. The skewness balancing algorithm used for filtering was adopted as a new method for obtaining an optimal intensity threshold such that the “pure” road points can be labeled. The rotating neighborhood algorithm was developed to remove narrow roads, such as corridors leading to parking lots or sidewalks. The hierarchical fusion and optimization algorithm was applied to extract the complete urban road network. Husain and Vaishya [22] proposed a new pipeline for the detection of road surface and corresponding centerline and boundary lines using terrestrial LiDAR data. The pipeline includes two basic phases. In the first phase, the candidate road surface pixels of intensity image converted from LiDAR point clouds were detected after applying an averaging mask. In the second phase, centerlines and boundary lines were detected by forming vertical grids at detected road surfaces. Chen et al. [23] developed a higher-order tensor voting-based method, which could locate road junctions by identifying multi-directional features in an encoded high-order tensor model.

Although many efforts have been made by previous studies, the problem of road detection and extraction from LiDAR data is still far from being solved. A common framework for extracting roads from airborne LiDAR point clouds mainly includes two phrases: (1) distinguishing ground points from non-ground points, referred to as filtering, which can be achieved by elevation analysis, and (2) extracting road points from ground points. The ground points typically contain roads, grass, bare land, parking lots, etc. The elevation is not a suitable feature for further point classification for road extraction from ground points, because different neighboring points on the ground may have similar elevations. Therefore, some studies [13–16,20] use the intensity of LiDAR data as the main cue to distinguish roads from other ground points. However, the intensity value of LiDAR data is affected by various factors such as surface reflectance, atmospheric attenuation,

incident angle, etc., and cannot describe the texture and structure of ground objects fully, making it difficult to extract high-quality road footprints from ground points even with the application of LiDAR intensity. In addition, point clouds usually lack spectral information, which poses difficulties for reliable object recognition. Moreover, because of the irregular distribution of point clouds, more effort is needed to extract accurate break lines or features, such as road edges. Due to the aforementioned problems of road extraction from the single dataset of point clouds, some studies have started to focus on using the combination of optical images and point clouds for road extraction.

A current airborne LiDAR system usually integrates a high-resolution charge coupled device (CCD) metric camera, from which high-resolution aerial images can be collected along with laser scanning data. Optical images can provide spectral and texture information simultaneously, which can be fused with LiDAR point clouds to identify ground objects, including mandate ones such as roads. Zhu et al. [24] presented an innovative automatic road extraction technique that combined information from digital images and laser scanning data, which could recover the hidden road edges in digital images by removing any high objects identified by point clouds. Hu et al. [25] focused on the integrated processing of high-resolution image and LiDAR data for automatic extraction of grid structured urban road network. The method firstly segmented the road areas from the LiDAR data using the intensity and height information. Then, road detection was performed by iterative Hough transform based on the contextual targets (i.e., parking lots and grasslands) obtained by image analysis. Youn et al. [26] presented a method for automatically extracting urban road networks from orthophotos and LiDAR data. The method started from the subdivision of a study area into small regions based on the homogeneity of the dominant road directions from the orthophotos. A process called “acupuncture” was used to select the road candidates in each region. Finally, extracted road candidates were edited to avoid collocation with non-road features such as buildings and grass fields. Wang et al. [27] preprocessed both point clouds and aerial images, and then used an improved mean shift algorithm to classify the LiDAR data fused with spectrum attributes into clusters, such as roads, buildings, vegetation, etc. Sameen and Pradhan [28] proposed a two-stage optimization strategy based on fuzzy object analysis for extracting urban road networks from LiDAR data combined with spectral information. In the strategy, the scale parameter was optimized using the Bhattacharyya distance, while the shape and compactness parameters were optimized using the Taguchi method. Milan [29] developed an integrated object-based analysis framework for detecting and extracting various types of urban roads from high-resolution optical images and LiDAR data.

Nevertheless, the problem of road detection and extraction from the combination of point clouds and optical images still poses challenges to the research community: (1) The accuracy of road extraction results in complex urban scenes is low. Although the fusion of point clouds and optical images can improve the quality of road extraction, a large number of high buildings, tree shadows, vehicles, and road traffic signs reduce the integrity of road data and affect the extraction of road features. (2) The current mainstream road points extraction methods mainly adopt intensity and elevation of point clouds as the input features to a classifier. However, even if these two features are used [16,27], they cannot fully describe the difference between road areas and non-road areas (such as parking lots, low grass, bare land, etc.), resulting in low quality of the extracted roads.

As the development of deep learning and its applications to remote sensing data processing progresses, it shows that deep learning networks can learn the deep features of input data [30–35], which can be used as compensation for artificially constructed features to achieve high-accuracy classification and segmentation of remotely sensed data, including point clouds. Maturana and Scherer [31] proposed a method called VoxNet, which converted point clouds into voxels, and successfully applied 3-D convolutional neural networks (CNN) to process voxel data, achieving the recognition of geometric shapes in point clouds. Sun et al. [32] converted the point clouds into a normalized digital surface model (NDSM) image, then combined it with the spectral image, and adopted a

classic autoencoder to semantically segment the point clouds. However, the first step of each above-mentioned deep learning model is to rasterize the point clouds by interpolation. PointNet [33] is the first deep learning model that is directly applicable to point clouds. It was designed to be invariant to the permutations of input points; thereby, the process of classifying points into specific categories was greatly simplified into an end-to-end method, without the need for preprocessing steps such as rasterization. PointNet could learn global features from a set of point data, but it lost the description of local geometric features. To solve this problem, the authors further proposed a hierarchical network PointNet++ [34] to capture fine geometric structures from the neighborhood of each point. Following these two models, several modifications and variations were developed. Inspired by the scale invariance feature transform (SIFT) feature descriptor, PointSIFT [35] added a series of orientation-encoding units in the architecture of the PointNet++, to combine features from different orientations. Because the simple max or average pooling used in PointNet++ did not make the best use of local contextual information, PointWeb [36] proposed an adaptive feature adjustment module by which each feature could be pulled or pushed by the other features in a given local region according to the adaptively-learned impact indicators. Local contextual information could be encoded in this manner and the experimental results outperformed the compared ones. Liu et al. [37] proposed a point context encoding (PointCE) module to learn a set of scaling factors, by which the network could selectively focus on a few important intermediate features. Similarly, LAE-Conv [38] learned the coefficients that represented the importance of a neighborhood to the central point by multilayer perceptron (MLP), then aggregated the central point features as a weighted sum of its neighbors' features. Hu et al. [39] proposed an efficient and lightweight network called RandLA-Net for large-scale point clouds segmentation. This network utilized random point sampling to achieve remarkably high efficiency in terms of memory and computation. A local feature aggregation module was proposed to capture and preserve geometric details by progressively increasing the receptive field for each 3D point. The experiments showed that RandLA-Net could process 1 million points in a single pass up to  $200\times$  faster than existing approaches.

Compared with the extensive application of deep learning to terrestrial laser scanning (TLS) and mobile laser scanning (MLS) point clouds, there are relatively fewer studies on airborne laser scanning (ALS) data. Ödemir et al. [40] classified aerial point clouds into ground-level objects, including roads and pavements, vegetation, buildings' facades and roofs, and three deep learning algorithms and one other machine learning algorithm were tested and evaluated. Wen et al. [41] presented a global-local graph attention convolution neural network (GACNN) that combined edge attention and density attention, which was applied to the classification of unstructured 3D point clouds obtained by airborne LiDAR and which achieved good results. However, in these applications, road and other ground-level objects, such as pavements and parking lots, belong to the same class of impervious surfaces. Bearing all the abovementioned in mind, the purpose of this article is to introduce a new urban road extraction framework from airborne LiDAR point clouds and high-resolution co-registered images. The framework consists of three phases:

- (1) Road points extraction by PointNet++, where the features of the input data include not only those selected from raw LiDAR points, such as 3D coordinate values, intensity, etc., but also the DN values of co-registered images and generated geometric features to describe a strip-like road.
- (2) Two-step post-processing of the extracted road points from PointNet++ by graph-cut and constrained triangulation irregular networks (CTINs) to smooth road points and to remove clustered non-road points.
- (3) Centerline extraction based on a modified method originally proposed in [21], where road segment gaps and holes caused by the occlusion of vehicles, trees, and buildings are recovered by estimating the connection probability of the road segment through collinearity and width similarity.

In summary, there are two main contributions of the work:



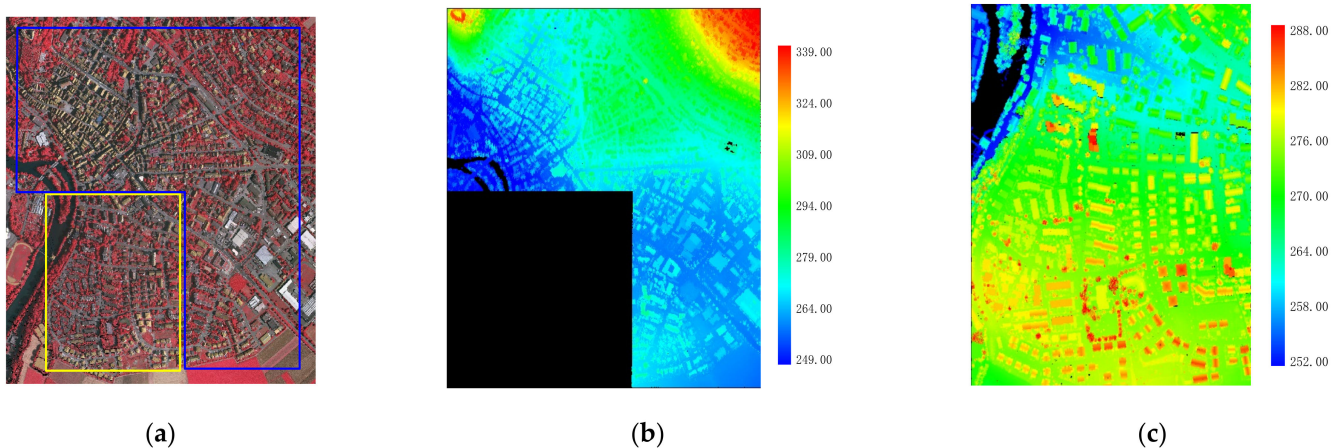
- (1) Strip-like descriptors are proposed to characterize the geometric property of a road, which can be applied to both point clouds and optical images. Strip-like descriptors, together with other features, are input to the PointNet++ network, resulting in a more accurate road point classification.
- (2) Collinearity and width similarity are proposed to estimate the connection probability of road segments to repair the road gaps and holes caused by the occlusion of vehicles, trees, buildings, and some omission errors.

Experimental results show that the proposed framework not only outperformed previous methods [14,21,42] in terms of completeness and correctness, but also extracted narrower residential streets, which have normally been neglected by previous studies.

## 2. Materials and Method

### 2.1. Sample Materials

Experiments were conducted on the Vaihingen dataset provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) to compare the results of the proposed framework with previous methods. The data were acquired in August 2008 by Leica ALS50 with an average flying height of 500 m and a 45° field of view. The point density was approximately 4.0 points/m<sup>2</sup>. In this dataset, all LiDAR points contain the x-, y-, z-coordinates in the WGS84 reference coordinate system, intensity values, and other attributes that were not likely to be useful for classification. In addition, orthophotos that had been co-registered with the point clouds could provide DN values of R, G, B channels for the corresponding laser scanning point. Ground truth was manually classified as roads, public patches, buildings, and low, medium, and tall vegetations. The testing area was located in the center of the city of Vaihingen, with an area of 500 m × 600 m. The testing area included diverse landcover, mainly roads, buildings, trees, lawns, vehicles, a river, etc., which was an adequately complicated landscape for road detection in practice. The training data contained about 3,474,448 points covering an area of 1000 m × 1100 m. The training and testing areas are illustrated in Figure 1.



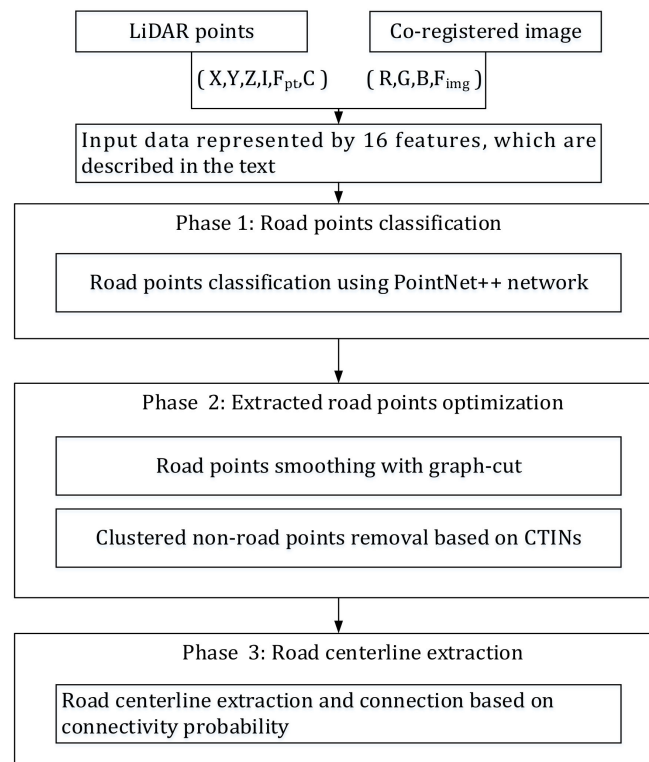
**Figure 1.** Top view of the Vaihingen dataset. (a) Optical image, (b) Zoomed in training, and (c) testing area. Training and testing area correspond to blue and yellow boxes in (a), respectively.

### 2.2. Method

#### 2.2.1. Overview of Method

The workflow of the proposed approach is described in Figure 2, which is broadly divided into three phases: road points classification, extracted road points optimization and road centerline extraction. In the first phase, the road points were extracted from the input data using PointNet++ network, which were termed as initial road points. The input data are those featured by three coordinate values of the point clouds, intensity, RGB DN values from the co-registered image, and nine geometric features that were generated

by the proposed approach to describe road geometry both in point clouds and image space. In the second phase, graph cut and non-road cluster filtering based on CTINs were performed to refine the initial road classification results. In the third phase, collinearity and width similarity were proposed to estimate the connection probability of the road segments, thereby improving the connectivity and completeness of the road network. The experimental results were evaluated on different quantitative metrics.



**Figure 2.** Workflow of the proposed framework.

### 2.2.2. Road Points Classification

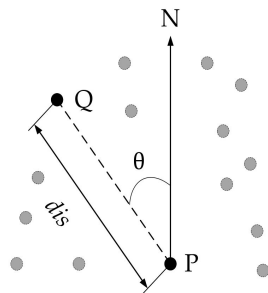
As mentioned in the workflow of the proposed approach, the initial road points were labeled by PointNet++. A crucial point of PointNet++ is to learn multiple layers of local context information progressively. First, the raw dataset was grouped into a set of overlapping sub-datasets. Local features were learned from these sub-datasets to generate low-level features. Higher-level features were learned from these low-level ones, subsequently, with the same procedure. These grouping and learning processes were repeated until the global features of all input points were obtained. A hierarchical propagation strategy was adopted by propagating the features extracted from sub-dataset points back to the original dataset.

PointNet++ can recognize fine-grained patterns, which helped produce better results in complex scenes. Moreover, PointNet++ showed higher robustness to changes in point density. Therefore, we chose PointNet++ to label initial road points in the first phase.

However, as the original of PointNet++ is mainly for the segmentation of point clouds acquired by terrestrial LiDAR, the input data were merely the three-dimensional coordinate values of point clouds. Some studies [43,44] expanded the input features to include intensity and RGB color information acquired by fusing optical images or by multispectral LiDAR. Although the PointNet++ network claims it can provide many deep learning features during the learning process, these features are based on the entire dataset and are not always optimal for the extraction of specific objects, such as roads. This is mainly caused by the application of unsupervised learning in the model initialization stage, where MLP mapping

with shared weights showed difficulty in fully characterizing the spatial distribution of the point clouds, resulting in the network lacking the ability to learn geometric features of some specific objects, such as a strip-like road. Therefore, several additional features, such as intensity, RGB DN values, point density, and strip descriptors were used as input features to the PointNet++ model in the proposed framework. Most of the features are straightforward; two parameters to describe a strip-like road are expounded in the following subsections.

First, project the point clouds onto the x-y plane. Given a point P, which is termed a reference point, define a virtual point Q, which is a point *dis* far from point P and the direction to it is  $\theta$ , measured counterclockwise from the North, as shown in Figure 3. With the variance of *dis* and  $\theta$ , lots of virtual points can be defined. The virtual points which possess the same or similar intensity values are identified as similar virtual points.



**Figure 3.** Locations of a reference point and virtual points.

Length and width are used to characterize the elongation and narrowness of a road, both of which need a defined main direction. Given a reference point, count the number of consecutive similar virtual points in a given direction. Define the direction with the largest number of consecutive similar virtual points as the main direction of the reference point, as shown in Figure 4. It is worth noting that there may be several main directions of a given reference point, depending on the similarity of paved material in the neighborhood of the reference point. Armed with the main direction, the length of a road  $T_{pt\_length}$  can be defined by the number of consecutive similar virtual points along the main direction. In the case of multiple main directions,  $T_{pt\_length}$  is the same along each direction, and any one of these directions can be the main direction. Considering that road width varies in intersections, main direction divergence  $T_{pt\_div}$  is proposed to describe road width variation in road intersections. It is defined as the number of directions that are clockwise and counterclockwise rotated from the main direction, under the condition that, along each direction, the number of consecutive similar virtual points is less than the difference between  $T_{pt\_length}$  and a given threshold. Figure 5 shows the strip-like descriptors of point clouds in different regions. Calculation details are as follows:

- (1) Generate the virtual point set  $Q = \{q_i(x_i, y_i)\}$  of the current reference point  $p(x_0, y_0)$  by Formula (1), where  $dL = 2.5 \times \text{the maximum width of the roads}$ . The  $0^\circ$  direction is defined as the direction where P points to North. The increase of  $\theta$  by  $10^\circ$  can reduce computation load while ensuring the accuracy of strip descriptors [21].

$$\begin{cases} x_i = x_0 + dis * \cos(\theta) \\ y_i = y_0 + dis * \sin(\theta) \end{cases} \quad dis = 1, 2, \dots, dL; \theta = 0^\circ, 10^\circ, 20^\circ, \dots, 340^\circ, 350^\circ \quad (1)$$

- (2) Calculate the average intensity value of each virtual point. Select a virtual point  $q$  from the virtual point set  $Q = \{q_i(x_i, y_i)\}$ . Search the k-neighborhood points on the projected LiDAR points of  $q$ , and assign the average intensity of all points in the neighborhood to the intensity value of  $q$ . To guarantee that each virtual point from the road contains at least one neighborhood point, the value of k is 2 times the average point spacing. In cases where a virtual point contains no neighborhood points, its intensity value is set to  $-1$ , indicating an invalid point.

- (3) Calculate the intensity difference between each virtual point and the reference point and mark the virtual points with an intensity difference lower than a given threshold  $T_I$  as similar virtual points. Count the number of consecutive similar virtual points starting from the reference point in 36 directions from  $0^\circ, 10^\circ, 20^\circ, \dots, 350^\circ$ , and denote them by  $\{N_0, N_1, \dots, N_{35}\}$ . This step can be formulated as follows:

$$\Delta I_{dis, \theta} = |I_p - I_{q(dis, \theta)}| \quad (2)$$

$$M_{dis, \theta} = \begin{cases} 1, & \text{if } \Delta I_{dis, \theta} < T_I \\ 0, & \text{if } \Delta I_{dis, \theta} > T_I \end{cases} \quad (3)$$

where  $I_p$  is the intensity value of the reference point and  $I_{q(dis, \theta)}$  is the intensity value of the virtual point at the distance  $dis$  in the  $\theta$  direction with regard to the reference point.  $T_I$  is the intensity difference threshold, and, as noted in [20], the range of road intensity difference in a local road area is around  $-15 \sim +15$  if the road is paved with the same material, so the absolute value of  $T_I$  is set to 15 for the data sets tested in this study. " $M_{dis, \theta} = 1$ " is used to mark the similar virtual points. Therefore, the number of consecutive similar virtual points in any direction can be counted by the following steps:

- Set the initial values of  $N_i$  ( $i = 0, 1, \dots, 35$ ) to 0.
  - Set the initial direction  $\omega$  to  $0^\circ$ , denote the virtual points in this direction by  $Q_\omega = \{q_1, q_2, \dots, q_L\}$ , where the points are sequenced by the increasing distance from the virtual points and the reference point.
  - For a given point  $q_i$  ( $i = 1, 2, \dots, dL$ ) from  $Q_\omega$ , if its corresponding  $M_{dis, \theta} = 1$ , then go to the next point  $q_{i+1}$  and add 1 to  $N_i$ . Otherwise, stop the counting for this direction, set  $\omega = \omega + 10^\circ$ , and go to step b.
  - Repeat the steps until  $\omega = 360^\circ$ .
- (4) Suppose  $\varepsilon$  is the main direction of P, which, according to the definition of the main direction, is the direction with the largest number of consecutive similar virtual points. Let  $N_\varepsilon$  be the number of continuous similar virtual points in the  $\varepsilon$  direction. Iteratively calculate the difference between  $N_\varepsilon$  and  $N_i$ , record them as  $\{n_0, n_1, \dots, n_{35}\}$ , which can be expressed as:

$$N_\varepsilon = \max(N_i), i = 0, 1, 2, \dots, 35 \quad (4)$$

$$n_i = N_\varepsilon - N_i, i = 0, 1, 2, \dots, 35 \quad (5)$$

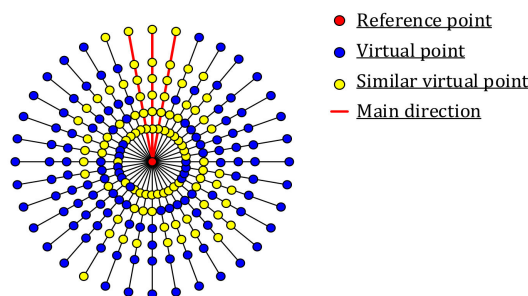
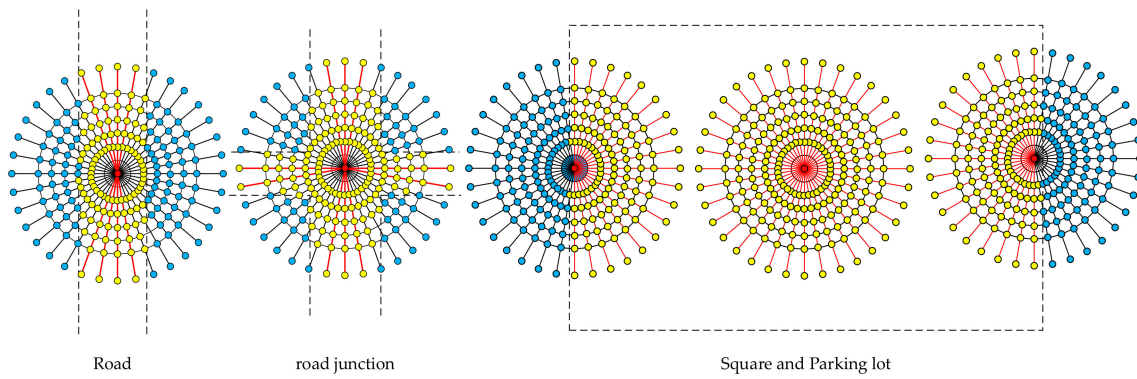


Figure 4. Main direction of reference point.





**Figure 5.** Strip-like descriptor of roads and squares (Red point: the reference point; Yellow points: virtual points with similar intensity value to the reference point; Blue points: other virtual points; Red lines: the main directions; Dotted line: delineates the area of roads, squares and parking lots).

The divergence of the main direction  $\varepsilon$  of the reference point can be obtained by calculating two parameters:  $M_{\varepsilon L}$  and  $M_{\varepsilon R}$ .  $M_{\varepsilon L}$  refers to the number of consecutive directions in which  $n_i$  is less than a given threshold  $T_{dL}$  in a counterclockwise direction starting from the main direction  $\varepsilon$ . While  $M_{\varepsilon R}$  refers to the number of consecutive directions in which  $n_i$  is less than a given threshold  $T_{dL}$  in a clockwise direction starting from the main direction  $\varepsilon$ .  $T_{dL}$  is the difference threshold. A too large value of  $T_{dL}$  causes roads with large widths to be labeled as parking lots or other patches of paved surface, while a too small value of  $T_{dL}$  will distinguish some narrow lanes through which a car can hardly pass. It is set to half of the width of the widest road in the area, which is obtained by trial-and-error. The detailed steps are as follows.

- a. Rotate from the main direction  $\varepsilon$  counterclockwise, every  $10^\circ$ , to check whether  $n_i$  is less than the given threshold  $T_{dL}$  where subscript  $i = (\frac{\varepsilon}{10^\circ} + 1)$  or  $i = 0$  if  $\varepsilon = 350^\circ$ . If it is true and  $i \neq \varepsilon/10^\circ$ , then,

$$i = \begin{cases} i + 1, & \text{others} \\ 0, & \text{if } (i + 1) = 36 \end{cases} \quad (6)$$

$$M_{\varepsilon L} = M_{\varepsilon L} + 1 \quad (7)$$

Otherwise, if  $n_i > T_{dL}$ , or  $i = \varepsilon/10^\circ$ , continue to the next step.

- b. Rotate from the main direction  $\varepsilon$  clockwise, every  $10^\circ$ , to check whether  $n_j$  is less than the given threshold  $T_{dL}$  where subscript  $j = (\frac{\varepsilon}{10^\circ} - 1)$  or  $j = 35$  if  $\varepsilon = 0^\circ$ . If it is true and  $j \neq \varepsilon$ , then,

$$j = \begin{cases} j - 1, & \text{others} \\ 35, & \text{if } (j - 1) = -1 \end{cases} \quad (8)$$

$$M_{\varepsilon R} = M_{\varepsilon R} + 1 \quad (9)$$

Otherwise, if  $n_j > T_{dL}$ , or  $j = \varepsilon/10^\circ$ , continue to the final step.

Finally,  $T_{pt\_length}$ ,  $T_{pt\_div\_e}$  and  $T_{pt\_div}$  are calculated by the following formulas:

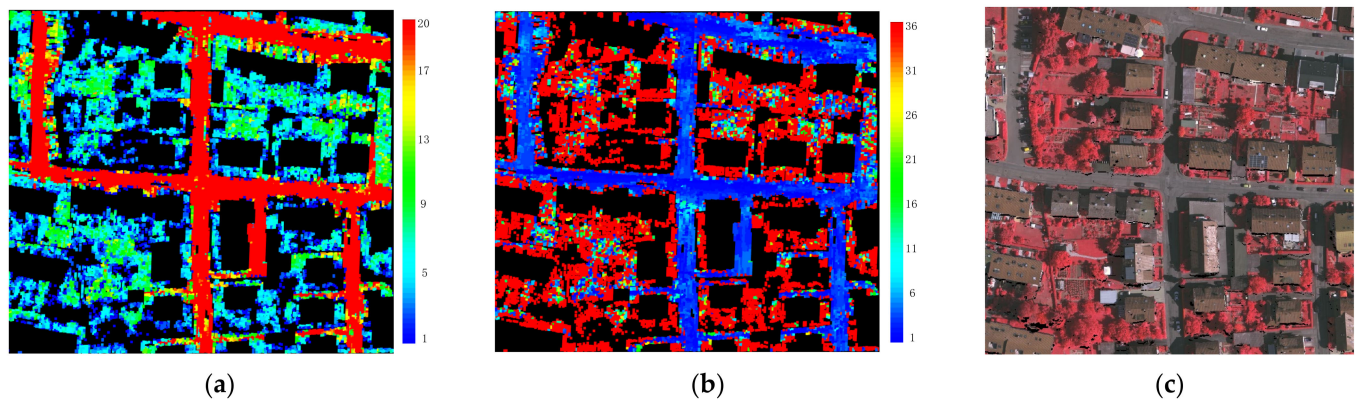
$$T_{pt\_length} = N_\varepsilon \quad (10)$$

$$T_{pt\_div\_e} = \begin{cases} M_{\varepsilon L} + M_{\varepsilon R}, & \text{others} \\ 36, & \text{if } (M_{\varepsilon L} + M_{\varepsilon R}) > 36 \end{cases} \quad (11)$$

$$T_{pt\_div} = \begin{cases} T_{pt\_div\_e} & \text{if } \varepsilon \text{ is the only main direction of the reference point} \\ \max(T_{pt\_div\_e i}) & \text{if } (i < 36) \text{ are the main directions of the reference point} \end{cases} \quad (12)$$

Denote the counterpart of  $T_{pt\_length}$  and  $T_{pt\_div}$  in the image space as  $T_{img\_*length}$  and  $T_{img\_*div}$  (where  $*$  collectively denotes  $R, G, B$ ). The calculation of these features is the

same as that of  $T_{pt\_length}$  and  $T_{pt\_div}$ , except that  $T_{img\_*length}$  and  $T_{img\_*div}$  are calculated from the image space. Reference point and virtual points are projected onto the co-registered image, followed by the steps described above, where laser points replaced by image pixels and channels R, G and B are viewed as 3 different intensity values; therefore, 6 features, namely  $T_{img\_Rlength}$ ,  $T_{img\_Rdiv}$ ,  $T_{img\_Glength}$ ,  $T_{img\_Gdiv}$ ,  $T_{img\_Blength}$  and  $T_{img\_Bdiv}$ , are formed. Figure 6 shows the strip descriptors conducted to the filtered point clouds, where only road points and other ground-level points remained. Table 1 summarizes the employed features used for the experiment.

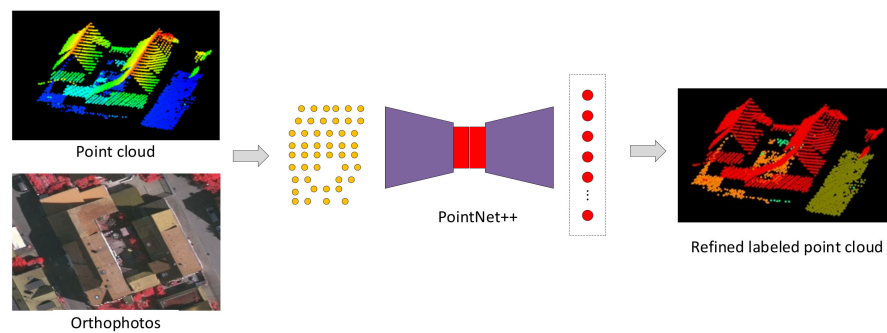


**Figure 6.** Strip descriptors of points. (a)  $T_{pt\_length}$ . (b)  $T_{ot\_div}$ . (c) Optical image.

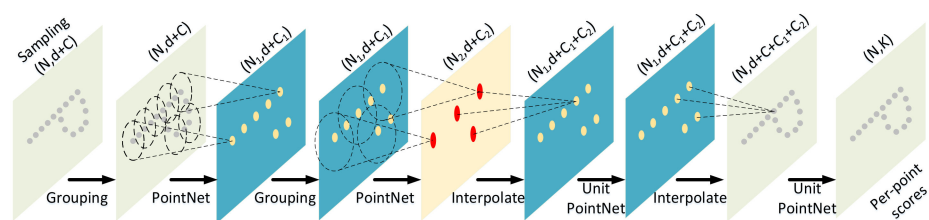
**Table 1.** Set of features used in the experiment.

Features	Symbol	Definition
Coordinate values	$x, y, z$	The coordinate values of a given point.
Intensity	$I$	The intensity of a given point.
Point density	$T_{pt\_density}$	The number of points per unit area.
Strip descriptors for point clouds	$T_{pt\_length}$ $T_{pt\_div}$	$T_{pt\_length}$ and $T_{pt\_div}$ are used to describe the length and main direction divergence (width) of a road in point clouds, see the text for details.
Color	$R, G, B$	The DN values of a co-registered image.
Strip descriptors for optical image	$T_{img\_Rlength}$ $T_{img\_Rdiv}$ $T_{img\_Glength}$ $T_{img\_Gdiv}$ $T_{img\_Blength}$ $T_{img\_Bdiv}$	$T_{img\_*length}$ and $T_{img\_*div}$ are used to describe the length and main direction divergence (width) of a road in an image, see the text for details.

Prior to the training of the PointNet++, point clouds of the training and testing data are divided into small 3D cubic voxels. The size of a voxel is 200 m. Each point is represented by a 16D vector where  $x$ -,  $y$ -,  $z$ - coordinates, intensity, density,  $T_{pt\_length}$  and  $T_{pt\_wid}$  are obtained from LiDAR point clouds, while digital numbers of R, G, B channels,  $T_{img\_*length}$  and  $T_{img\_*div}$  ( $* = R, G, B$ ) are provided by the co-registered image. Figures 7 and 8 show the schemes of the basic steps of using PointNet++ for point clouds classification.



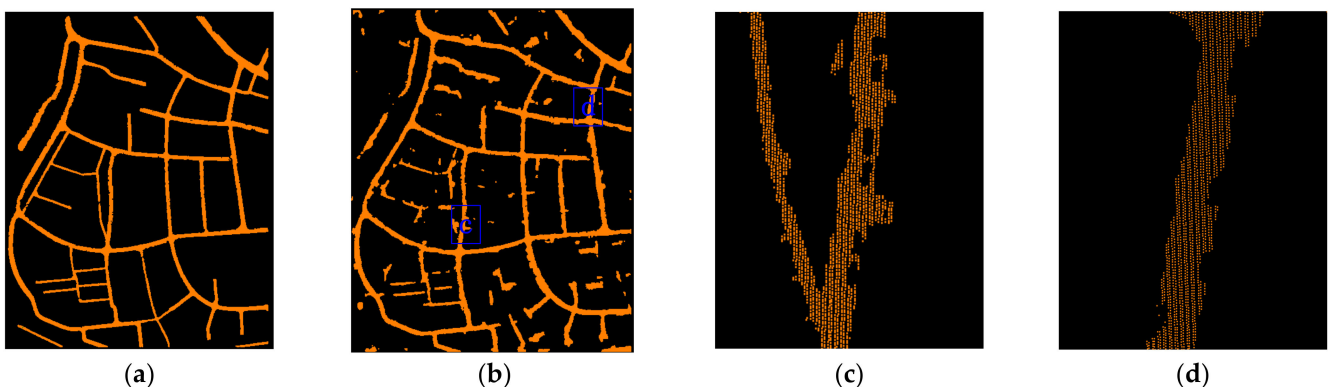
**Figure 7.** Initial road points classification using PointNet++.



**Figure 8.** Illustration of the deep feature learning architecture in PointNet++ [34].

In the training process, the Adam optimizer is used with an initial learning rate of 0.001, a momentum value of 0.1 and a batch size of 16. The learning rate is iteratively reduced based on the current epoch by a factor of 0.5. Moreover, the training process lasts a total of 300 epochs and the weights are saved if the loss function decreased. The network is implemented via TensorFlow and performed on a computer with NVIDIA Corporation GP102 (GeForce GTX 1080 Ti) 32 GB GPU. The code of PointNet++ can be downloaded online: [https://github.com/yanx27/Pointnet\\_Pointnet2\\_pytorch](https://github.com/yanx27/Pointnet_Pointnet2_pytorch) (accessed on 25 June 2021).

In the experiments, training samples and the standard testing data were obtained by hand labeling using the editing tools provided by LiDAR\_Suite, an airborne point clouds processing software developed by the author's R&D group that has been commercialized. Figure 9 shows the results of the initial road points extracted at this stage. There were many small clustered and isolated points (Figure 9b), indicating commission errors. Furthermore, the edges of the roads were not smooth, and there were holes inside the road caused by vehicles, trees, etc., as shown in Figure 9c,d.



**Figure 9.** Road points in the testing area. (a) The true road points are obtained by hand labeling. (b) The initial road points are extracted with geometric PointNet++. (c) & (d) The zoomed in displaying results of the selected area in (b).

### 2.2.3. Two-Step Post-Processing of Initial Road Points

The initial road points extracted by the first phase of the proposed workflow contained both commission and omission errors. Therefore, two-step post-processing by graph-cut and CTINs were applied to optimize the classification results.

#### Road Point Smoothing with Graph-Cut

As pointed out in [45], local smoothing of the initial classification results can effectively remove isolated “road points”, while the omitted road points located on the edge and inside the road will be relabeled. A graph-cut-based mathematical framework was proposed in [45] for this purpose. As illustrated in Figure 10, this approach can be divided into the following three steps: construction of the graphical model, construction of the energy function, and redefinition of point labels. The model was constructed by the undirected adjacency graph, expressed as  $\{V, E\}$ , where  $V$  denotes the nodes (representing the initial road points) and  $E$  denotes a set of edges connecting the nodes. Unlike [45], we did not directly use a k-nearest neighbors (KNN) algorithm to search for nodes; rather, we pre-defined a flattened cuboid as the neighborhood of a given point whose height is much smaller than its length and width, as shown in Figure 11, to prevent the points reflected from low objects, such as bushes, along the road from the construction of the graph, because a road surface can be viewed as a flat plane along its direction. The interested reader is referred to the cited reference for a detailed description of the method.

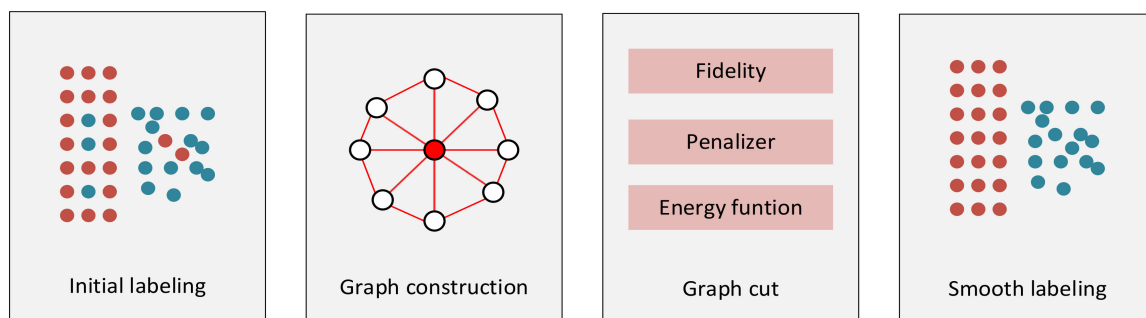


Figure 10. Road point smoothing via graph-cut [45].

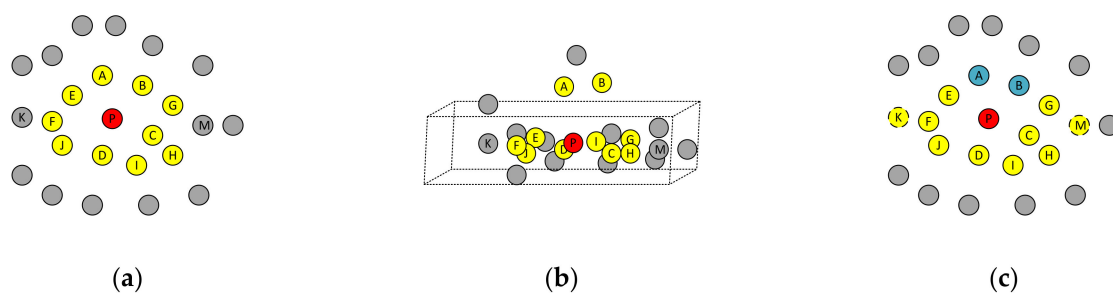


Figure 11. Illustration of nodes selection. (a) Nodes selected by KNN algorithm; (b) Flattened cuboid; (c) Nodes selected by the improved method.

#### Clustered Non-Road Points Removal Based on CTINs

Though some commission and omission errors can be removed after the graph-cut smoothing, small clusters of non-road points can remain. CTINs is applied to eliminate these clusters. The mechanism behind CTINs is based on the following observation: The side length of a triangle formed by road points is smaller than the side length of a triangle formed by the edge points of the road and the adjacent non-road points. Therefore, side length can be used as the constraint to divide the triangles constructed by Delaunay triangulation performed on the smoothed initial road points into an unconnected cluster



of triangles and isolated points. Since a road surface is well connected, and it extends for several or even several tens of kilometers in urban areas, the area of the road surface is normally larger than many clusters of non-road points, which can be removed by setting a coverage area threshold. For the elimination of other large area clusters composed of non-road points, such as parking lots, public parks, etc., we utilized the following facts: (1) The shape of a road is a regular rectangle or curved rectangle, where the length is far greater than the width, as shown in Figure 12a. (2) For a given urban area, most are occupied by residential and commercial buildings, which means the total area of roads is a small fraction of the total area of the given region, as shown in Figure 12b. (3) Clusters of non-road points from patches such as parking lots, public parks, etc., stretch omnidirectionally, so that the length and width of a bounding rectangle for the patch holds the same level of values, as shown in Figure 12c. Zhang et al. [46] pointed out that parameters describing object shape, such as length, width, etc., could effectively detect man-made objects. Therefore, it is possible to filter out large-area non-road point clusters, such as parking lots, by determining the ratio of length to width (aspect ratio) and point coverage of the bounding rectangle of such a cluster. The specific steps are as follows:

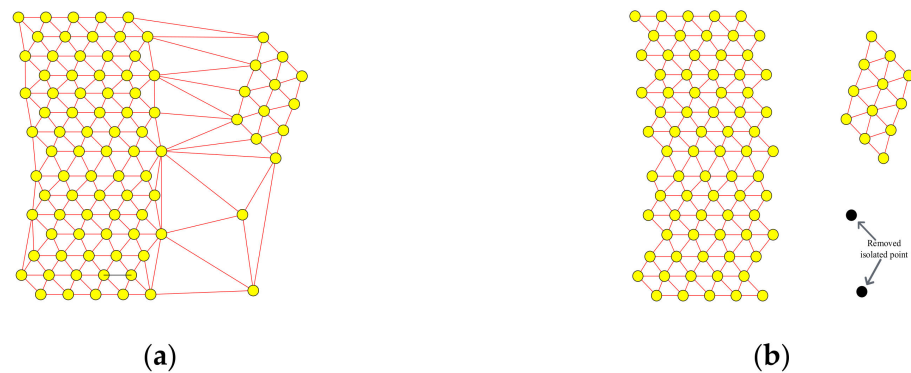
- (1) The Bower–Watson algorithm [47] is applied to construct the Delaunay triangulation of the graph-cut processed initial road points.
- (2) Traverse the edges of the triangles and remove the edges with lengths greater than  $T_L$ , which is set to twice the mean average point spacing in the raw point clouds. Figure 13 shows the triangles before and after this step is performed.
- (3) Remove isolated points, resulting in separated clusters of triangles, as shown in Figure 13b.
- (4) Calculate the total area of individual clusters of triangles obtained in step 3. Remove those clusters with areas less than a given threshold  $T_S$ , which is the minimum area of a road in the urban setting. The empirical value of  $T_S$  for urban areas is  $100 \text{ m}^2$ . The total area of clustered triangles can be calculated as follows: the points constructed from the triangles are projected onto the x-y plane. Their maximum and minimum X, Y values, which are denoted by  $X_{max}$ ,  $X_{min}$ ,  $Y_{max}$  and  $Y_{min}$ , respectively, are obtained. A rectangle can be formed by them. Grid the rectangle with the cell size of  $1.5 \times$  average point spacing of the raw dataset. Count all cells with at least one point inside and then sum up the area of these cells, which is the approximation of the total area of the clustered triangles. We used this approximation method rather than calculating the area of individual triangles, and then summed them up to reduce computational load.
- (5) For the remaining clusters of triangles, the minimum bounding rectangle (MBR) method proposed in [48] was used to form a bounding rectangle for each cluster. Denote the length, width and area of the rectangle by  $L_{MBR}$ ,  $W_{MBR}$  and  $S_{MBR}$ . Calculate aspect ratio  $lw_{CDT}$  and point coverage  $pc_{CDT}$  as follows:

$$\begin{cases} lw_{CDT} = \frac{L_{MBR}}{W_{MBR}} \\ pc_{CDT} = \frac{S_{CDT}}{S_{MBR}} = \frac{S_{CDT}}{L_{MBR} \times W_{MBR}} \end{cases} \quad (13)$$

where  $S_{CDT}$  is the area of a given triangle cluster calculated in step 4. If  $lw_{CDT} > T_{lw}$  and  $pc_{CDT} > T_{pc}$ , where  $T_{lw}$ ,  $T_{pc}$  are two predefined thresholds, then remove the cluster and all points inside. The values of  $T_{lw}$  and  $T_{pc}$  are 6 and 0.3, respectively, which are based on prior knowledge acquired by inspecting these parameters in a given urban region.

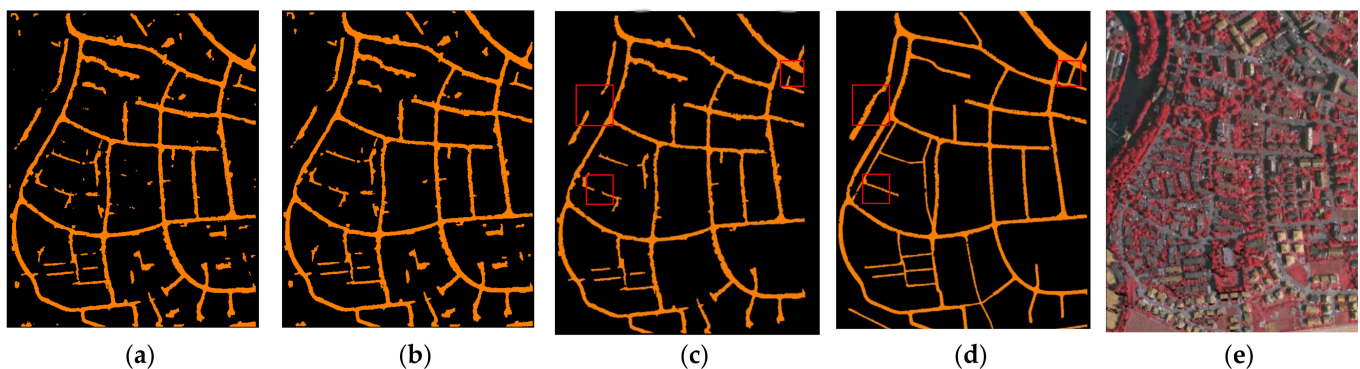


**Figure 12.** Illustration of the shape difference between a road and a parking lot. (a) Local Road; (b) Road Network; (c) Parking Lot.



**Figure 13.** Triangulated irregular network (TIN). (a) TIN formed by Delaunay triangulation; (b) Result of (a) after step 2 described in the text is performed.

The results of initial road points followed by the two-step post-processing are illustrated in Figure 14. It is obvious that most of the patches can be removed; however, gaps in a continuous road remain, as shown by the rectangle in Figure 14c,d which will be connected in the third phase of the proposed framework for centerline extraction.



**Figure 14.** Road point extraction results. (a) Road points extracted by PointNet++ with geometric features. (b) Smoothing with graph-cut. (c) Clustered non-road points removal. (d) Road points obtained by hand editing. (e) Optical image of the corresponding area.

### 2.2.3.3. Road Centerlines Extraction

Road centerlines are not only functional for the extraction of a complete road network, but also for quantitative comparison with previous studies. We mainly adopted the approach presented in Hui et al. [21], which is a hierarchical fusion and optimization

grouping strategy for road centerline extraction, for this purpose. In this approach, several linear structural elements (SE) of different lengths are used to yield different road extraction outcomes of different levels, and the results of each level are combined and optimized with those of the previous level in a bottom-up manner. However, the method cannot effectively solve the road interruption caused by the occlusion of vehicles, trees or buildings, resulting in poor connectivity of the extracted centerlines; this inconsistent topological relationship makes it difficult to form a complete road network. Therefore, this paper proposes the connection probability of two disconnected road segments as the criteria to determine if the segments should be connected. The main principles of the method are that the same road segment usually has the same width, and most roads are straight. Collinearity and width similarity are defined first to calculate the connection probability of road segments. The calculations of the two parameters are as follows:

- (1) First, define the connection range. To prevent connecting two lines that are far apart, a radius  $R$  is defined to determine the connection range. If the distance between the endpoints of two road segments is greater than  $R$ , the connection probability of the two lines sets to 0. The value of  $R$  is set to the width of the smallest residential block in the study area, which is 50 m in the testing region.
- (2) Collinearity ( $C_{line}$ ): A road with enough length may be curve in some part, which can change the extension direction of the road at the endpoints. To keep the true extension direction, a segment with length  $L$  at the end part of a road is selected to fit a straight line, as shown in Figure 15. A too-small value of  $L$  causes a large deviation between the direction of the fitted straight line and the actual direction, while a too-large  $L$  value may include the curved part of the road in the straight fitted line. It was set to 20 m in the study, by trial-and-error, which may be set as a constant parameter if similar studies are performed. Then, collinearity is defined by the following formula:

$$C_{line} = 1 - 0.5 \times \frac{d_1 + d_2}{L_1 + L_2} - 0.5 \times \frac{\theta}{90^\circ} \quad (14)$$

where,  $d_1$  and  $d_2$  are the distances from one end of a road to another fitted straight line (see Figure 15).  $\theta$  is the angle between the two fitted straight lines.  $L_1$  and  $L_2$  are the lengths of the matched lines. A matched line is defined as follows: create the buffer zone of a fitted straight line. Project the road segment surrounded by the buffer zone onto the direction of the fitted straight line, then  $L_i$  ( $i = 1, 2$ ) is the length of the projected segment. The buffer width is half the largest road width in the studied area.

- (3) Width similarity  $C_{width}$  is another parameter for calculating connection probability, which is defined by the following:

$$C_{width} = 1 - \frac{|w_1 - w_2|}{\text{Max}(dw)} \quad (15)$$

where,  $w_1$  and  $w_2$  are the widths of the two road segments.  $\text{Max}(dw)$  is the difference between the possible largest and smallest widths of roads in study area.

- (4) Connection probability of the two road segments can then be calculated by the following formula:

$$p = g_1 \times C_{line} + g_2 \times C_{wid} \quad (16)$$

where,  $g_1$  and  $g_2$  are the weights for collinearity measure and width similarity, respectively, both of which can be set to 0.5 if the two parameters are of similar importance. If the connection probability is greater than a given threshold  $T_p$ , then the two road segments are connected.

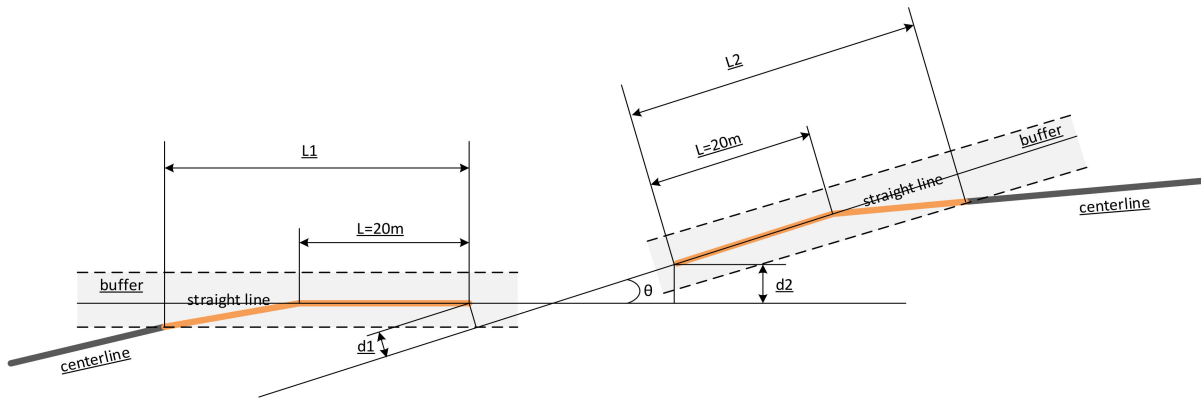


Figure 15. Collinearity measurement.

Figure 16 shows the results of the road centerlines extraction by [21] and by the proposed method, as well as the true centerlines determined by hand editing. Interruptions seen in Figure 16b have all been connected by the proposed method, as shown in Figure 16c.

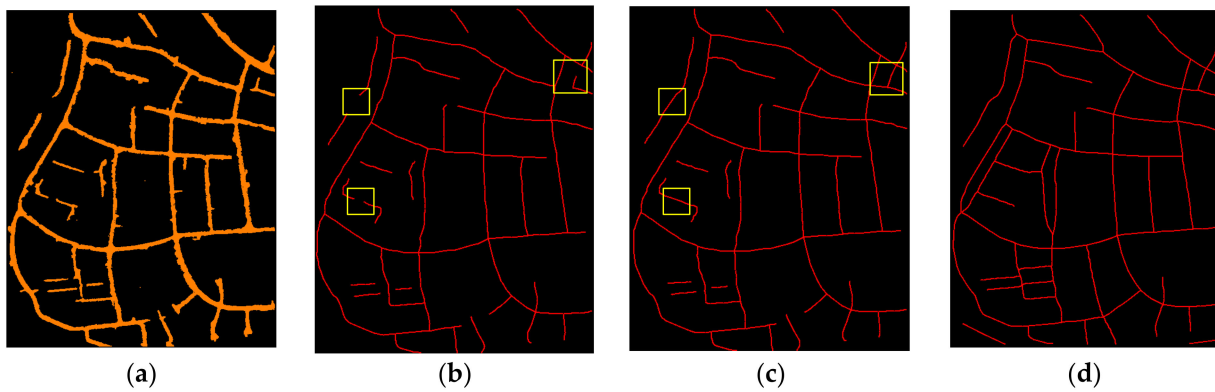


Figure 16. Centerline extraction results. (a) Road points after two-step post-processing; (b) Centerlines extracted by [21]; (c) Centerlines extracted by the proposed method; (d) True centerlines obtained by hand editing.

### 2.3. Evaluation Metrics

Three standard evaluation metrics, completeness ( $E_C$ ), correctness ( $E_{CR}$ ) and quality ( $Q$ ) [49,50] are adopted to quantitatively assess intermediate and final results:

$$E_C = \frac{TP}{TP + FN} \times 100\% \quad (17)$$

$$E_{CR} = \frac{TP}{TP + FP} \times 100\% \quad (18)$$

$$Q = \frac{TP}{TP + FN + FP} \times 100\% \quad (19)$$

where the true positive (TP) and the false positive (FP) are, respectively, the correctly and wrongly extracted road points, and the false negative (FN) refers to the missed road points.

For road centerlines, if the deviation of the extracted centerline from the true one (which is extracted manually) is less than the average point space, then it is a true positive centerline, otherwise, it is a false positive one.

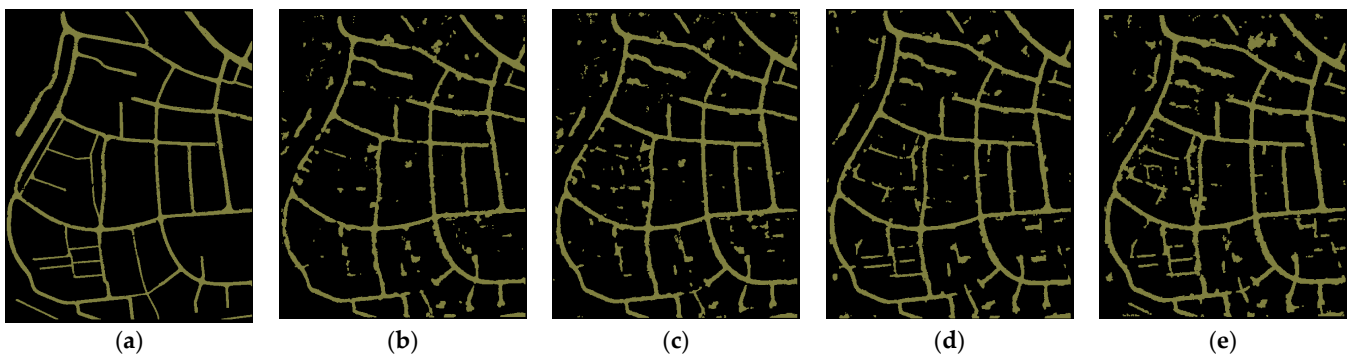


### 3. Results

As stated in the flowchart, the first phase of the proposed framework is to distinguish road from non-road points by PointNet++. To verify the effectiveness of PointNet++ for road points labeling, three experiments were conducted:

- (1) Input the raw point clouds directly to PointNet++ with each point represented by a 16D vector, described in Section 2. Six output classes are set to the PointNet++, namely, roads, public patches, buildings, and low, medium and tall vegetations. The result is denoted by  $M_{class6}$ .
- (2) The same input as (1) but only two outputs are set to the classifier: road points and non-road points; the result is denoted by  $M_{class2}$ .
- (3) Raw point clouds are firstly filtered by auto-adaptive progressive TIN proposed in [51] to obtain ground points, then input into PointNet++ to distinguish road and non-road points with and without the 9 strip descriptors, respectively; results are denoted by  $M_{class2\_from\_ground\_points}$  and  $M_{without\_geometric\_features}$ .

Figure 17 shows all the experimental results. In addition, the minimal width of roads is set to 2.0 m, through which most cars can pass.



**Figure 17.** Road point extraction results. (a) True road obtained by hand editing; (b) PointNet++ result with 6 output classes. Only road points are displayed; (c) PointNet++ result with only two output classes: road and non-road points. Only road points are displayed; (d) Input data to PointNet++ are firstly filtered with TIN, then are classified to two classes (road and non-road) with 16D vector representing the input points. (e) Input data to PointNet++ are firstly filtered with TIN then are classified to two classes (road and non-road), but 9 strip descriptors are dropped from the vector for representing input points.

Table 2 shows the quantitative assessment of the results from PointNet++ with different inputs, as well as the results after two-step post-processing. It can be seen from Table 2 that the PointNet++ classification result with the input of ground points obtained by auto-adaptive progressive TIN filtering the raw data, and with the 16D vector representation of the input points, achieved the highest accuracy, both in terms of completeness and quality. Correctness is the lowest, but it is still at the same level as  $M_{class6}$ ,  $M_{class2}$  and  $M_{class\_without\_geometric\_features}$ . It is obvious from Figure 17b–e that there are many patches of road points that actually belong to parking lots, public parks, etc. Furthermore, pepper-and-salt noises are common. Therefore, two-step post-processing aiming to improve the accuracy of the initial road points is necessary; the initial road points with the highest road integrity  $M_{class2\_from\_ground\_points}$  were selected for the two-step post-processing.

Results of two-step post-processing are shown in Figure 18, where Figure 18a illustrates isolated points elimination by graph-cut, and Figure 18b displays the final result after clustered non-road points removal by CTINs. The last two rows in Table 2 show that the improvement obtained by the three evaluation metrics is slight after the first post-processing step, but that they are improved significantly after the second post-processing step, indicating that, in general, patches such as parking lots, public parks, etc., affect the

road extraction more than other factors such as vegetations, vehicles, and buildings on or along a road.



**Figure 18.** Road point extraction results after two-step post-processing. (a) Smoothing the road points of Figure 17d with graph-cut; (b) Clustered non-road points removal from Figure 18a based on CTINs.

**Table 2.** Evaluation of road point classification results.

		$E_C$ (%)	$E_{CR}$ (%)	$Q$ (%)
PointNet++	$M_{class6}$	76.2	74.0	60.1
	$M_{class2}$	79.8	73.9	62.3
	$M_{class2\_from\_ground\_points}$	85.5	73.0	64.9
	$M_{without\_geometric\_features}$	82.7	71.0	61.9
Graph-cut smoothing		85.6	73.0	65.0
Clustered non-road points removal		84.7	79.7	69.6

Centerlines that are extracted in the final stage of the proposed framework not only serve for the generation of the road network, but also allow for further comparison with previous studies, as shown in Figure 19 and Table 3. The completeness, accuracy and quality achieved by the proposed approach are 97.0%, 86.3% and 84.1%, respectively. They are much higher than others, which the authors attribute to the following three reasons:

- (1) The deep features learned by PointNet++ and the geometric features generated in the first phase of the framework jointly improved the completeness of the extracted road points.
- (2) Two-step post-processing is applied to decrease both omission and commission errors in the initial road points distinguished by PointNet++.
- (3) Collinearity and width similarity are introduced to calculate the connection probability to further improve the completeness of extracted roads.

**Table 3.** Three evaluation metrics comparing the proposed method to previous studies. The indicators of PCD [14] are provided by [42].

Method	$E_{CR}$ (%)	$E_C$ (%)	$Q$ (%)
PCD [14]	53.2	58.3	38.5
SRH [21]	91.4	80.4	74.8
MTH [42]	73.8	53.4	44.9
Proposed method	<b>97.0</b>	<b>86.3</b>	<b>84.1</b>



**Figure 19.** Centerline extraction result. (a) True centerlines obtained from Figure 17a by hand editing; (b) Duplicate of Figure 18b for comparison; (c) Centerlines extracted from Figure 19b by the proposed framework.

#### 4. Discussion

Though the experimental results show that the proposed framework outperformed the other methods compared, a significant point that is worthy of further study is that the framework adopted many features other than 3D coordinate values of point clouds to the deep learning model, such as intensity, DN values from optical images, point density and strip descriptors. This motivated us to investigate more specific features describing the characteristics of roads and more intermediate features generated in the learning stage, so that higher quality road footprints could be extracted in the first phase of the framework. The quality of the final road network is mainly determined by the completeness of the extracted road centerlines. A reasonable connection between the endpoint of a road and a road intersection can improve the completeness of the centerlines, but how to measure the reasonability remains a challenge.

KNN search and fixed radius searches were used in the experiment when features were generated. Though they performed well, both of them possess drawbacks:

- (1) When the KNN search is applied to filter point clouds where only ground points (including road footprints) remained, the distance between a pair of points in the search region may be too large, resulting in useless features. That is, though the features can be generated, they are insignificant to the classifier or even decrease the classification accuracy.
- (2) When a fixed radius search is applied, some points may lack enough neighbor points for feature calculation, causing some feature loss.

In the selection of different deep learning models for the first step of the proposed framework, we found that the recently proposed model RandLA-Net takes similar training time as PointNet++, but, as for the efficiency of the application of the learned model to testing data, the former is much higher than the latter, though the classification accuracy of PointNet++ is slightly higher than RandLA-Net. The higher efficiency of RandLA-Net can be attributed to its random sampling mechanism. Considering that deep learning classification was the first step in our approach, which was followed by refined steps and the slightly higher classification achieved by PointNet++ in the experiment, we selected PointNet++ as the deep learning classifier in the proposed framework. However, it is worth noting that RandLA-Net can be a better choice in terms of its high efficiency in the model application. Further studies should be conducted on this issue.

Another observation that is interesting enough to study is the relationship between point density and road detectability. A road that can be detected and extracted from point clouds is strongly correlated to the point density of the dataset. Narrower roads or strip-shaped objects can be distinguished from higher density point clouds. To analyze the influence of point density on the extracted road width, the raw point clouds are resampled

to certain average point spacing to form 3 datasets with a point density of 4.0 points/m<sup>2</sup>, 1.0 points/m<sup>2</sup>, and 0.25 points/m<sup>2</sup>, respectively. The proposed framework was applied to extract road points from these datasets. Figure 20 and Table 4 show the results. It is worth noting that, when the point density was greater than 1.0 points/m<sup>2</sup>, meaning there were at least three points on the cross section of a road with a width of 3 m, the reduction in the point density displayed little influence on the extraction of the minimum width road. However, when the point density continued decreasing, such as when the point density was 0.25 points/m<sup>2</sup>, many narrow roads were not detected and extracted, as shown by the rectangle in Figure 20c.



**Figure 20.** Road points extracted under different point densities. (a) True road points. (b) Density = 4 points/m<sup>2</sup>. (c) Density = 1.0 points/m<sup>2</sup>. (d) Density = 0.25 points/m<sup>2</sup>.

**Table 4.** Road results extracted under different point densities.

PD	4.0 Points/m <sup>2</sup>	1.0 Points/m <sup>2</sup>	0.25 Points/m <sup>2</sup>
RW	2.0m	3.0 m	5.0m

## 5. Conclusions

A novel framework for extracting urban roads from LiDAR point clouds was proposed. Firstly, the recently developed deep learning model, PointNet++, was used to directly distinguish road points from non-road points, where two geometric features describing a strip-like road were generated both from point clouds and optical images, and input to PointNet++ with other features of the raw point clouds. Then, a two-step post-processing algorithm composed of graph-cut smoothing and CTINs-based clustered non-road points removal was performed to refine the initially-labeled road points. Finally, a connection probability based on centerline tracking was proposed to generate road network. The effectiveness of the proposed framework was verified by the experimental results using the Vaihingen dataset. Quantitative evaluations show that the proposed method outperformed others in terms of higher evaluation metrics and narrow street detection.

Although the proposed method was tested here with the Vaihingen dataset for the purpose of comparison with other methods, where point density was about 4.0 points/m<sup>2</sup>, it should be applicable to higher point density datasets, as new generation airborne LiDAR, such as Geiger-mode LiDAR, single-photon LiDAR, etc., are available for acquiring point clouds with a density over 100 points/m<sup>2</sup>. More studies should be focused on the applicability of the proposed framework and strip-like descriptors characterizing road geometric properties in very high-density point clouds, which are the future research topics of the authors' group.

**Author Contributions:** Conceptualization, H.M. (Hongchao Ma) and H.M. (Haichi Ma); Data curation, L.Z.; Formal analysis, H.M. (Haichi Ma); Funding acquisition, H.M. (Haichi Ma); Investigation, H.M. (Haichi Ma). and W.L.; Methodology, H.M. (Haichi Ma); Project administration, H.M. (Haichi



Ma); Resources, H.M. (Haichi Ma); Software, H.M. (Haichi Ma) and K.L.; Supervision, H.M. (Haichi Ma); Validation, H.M., K.L. and L.Z.; Visualization, H.M. (Hongchao Ma); Writing—original draft, H.M. (Haichi Ma); Writing—review and editing, H.M. (Haichi Ma). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China (2018YFB0504500), National Natural Science Foundation of China (Grant numbers 41601504 and 61378078), and National High Resolution Earth Observation Foundation (11-Y20A12-9001-17/18 and 11-H37B02-9001-19/22).

**Acknowledgments:** The test datasets were provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) Working Group and OpenTopography. Authors would like to thank them and anonymous reviewers for their constructive comments for the manuscript.

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

## References

1. Kobler, A.; Pfeifer, N.; Ogrinc, P.; Todorovski, L.; Oštir, K.; Džeroski, S. Repetitive interpolation: A robust algorithm for DTM generation from aerial laser scanner data in forested terrain. *Remote Sens. Environ.* **2007**, *108*, 9–23. [\[CrossRef\]](#)
2. Polat, N.; Uysal, M. Investigating performance of airborne LiDAR data filtering algorithms for DTM generation. *Measurement* **2015**, *63*, 61–68. [\[CrossRef\]](#)
3. Ma, H.; Ma, H.; Liu, K.; Luo, W.; Zhang, L. Direct georeferencing for the images in an airborne LiDAR system by automatic boresight misalignments calibration. *Sensors* **2020**, *20*, 5056. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Meng, X.; Wang, L.; Currit, N. Morphology-based building detection from airborne LiDAR data. *Photogramm. Eng. Remote Sens.* **2009**, *75*, 437–442. [\[CrossRef\]](#)
5. Hamraz, H.; Jacobos, N.B.; Contreras, M.A.; Clark, C.H. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. *ISPRS J. Photogramm.* **2019**, *158*, 219–230. [\[CrossRef\]](#)
6. Axelsson, P. DEM generation from laser scanner data using adaptive TIN models. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 110–118.
7. Hatger, C.; Brenner, C. Extraction of road geometry parameters from laser scanning and existing databases. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2003**, *34*, 225–230.
8. Jia, J.; Sun, H.; Jiang, C.; Karila, K.; Karjalainen, M.; Ahokas, E.; Khoramshahi, E.; Hu, P.; Chen, C.; Xue, T.; et al. Review on active and passive remote sensing techniques for road extraction. *Remote Sens.* **2021**, *13*, 4235. [\[CrossRef\]](#)
9. Rottensteiner, F.; Clode, S. Building and road extraction by LiDAR and imagery. In *Topographic Laser Ranging and Scanning: Principles and Processing*; CRC Press: Boca Raton, FL, USA, 2008; pp. 445–478. [\[CrossRef\]](#)
10. Ma, H.; Zhou, W.; Zhang, L.; Wang, S. Decomposition of small-footprint full waveform LiDAR data based on generalized Gaussian model and grouping LM optimization. *Meas. Sci. and Technol.* **2017**, *28*, 1–9. [\[CrossRef\]](#)
11. Tolt, G.; Shimoni, M.; Ahlberg, J. A shadow detection method for remote sensing images using VHR hyperspectral and LiDAR data. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS, Vancouver, BC, Canada, 24–29 July 2011; pp. 4423–4426. [\[CrossRef\]](#)
12. Rieger, W.; Kerschner, M.; Reiter, T.; Rottensteiner, F. Roads and buildings from laser scanner data within a forest enterprise. *Int. Arch. Photogramm. Remote Sens.* **1999**, *32*, 642–649.
13. Clode, S.; Kootsookos, P.J.; Rottensteiner, F. The automatic extraction of roads from LiDAR data. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2004**, *35*, 231–237.
14. Clode, S.; Rottensteiner, F.; Kootsookos, P.; Zelniker, E. Detection and vectorization of roads from LiDAR data. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 517–535. [\[CrossRef\]](#)
15. Choi, Y.W.; Jang, Y.W.; Lee, H.J.; Cho, G.S. Three-dimensional LiDAR data classifying to extract road point in urban area. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 725–729. [\[CrossRef\]](#)
16. Samadzadegan, F.; Hahn, M.; Bigdeli, B. Automatic road extraction from LiDAR data based on classifier fusion. In Proceedings of the Joint Urban Remote Sensing Event (JURSE), Shanghai, China, 20–22 May 2009; pp. 1–6. [\[CrossRef\]](#)
17. Zhu, Q.; Mordohai, P. A minimum cover approach for extracting the road network from airborne LiDAR data. In Proceedings of the International Conference on Computer Vision (ICCV) Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1582–1589. [\[CrossRef\]](#)
18. Zhao, J.; You, S. Road network extraction from airborne LiDAR data using scene context. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) Workshops, Providence, RI, USA, 16–21 June 2012; pp. 9–16. [\[CrossRef\]](#)
19. Matkan, A.A.; Hajeb, M.; Sadeghian, S. Road extraction from LiDAR data using support vector machine classification. *Photogramm. Eng. Remote Sens.* **2014**, *80*, 409–422. [\[CrossRef\]](#)
20. Li, Y.; Yong, B.; Wu, H.; An, R.; Xu, H. Road detection from airborne LiDAR point clouds adaptive for variability of intensity data. *Optik* **2015**, *126*, 4292–4298. [\[CrossRef\]](#)

21. Hui, Z.; Hu, Y.; Jin, S.; Yevenyo, Y.Z. Road centerline extraction from airborne LiDAR point cloud based on hierarchical fusion and optimization. *ISPRS J. Photogramm.* **2016**, *118*, 22–36. [[CrossRef](#)]
22. Husain, A.; Vaishya, R.C. Road surface and its center line and boundary lines detection using terrestrial LiDAR data. *The Egypt. J. Remote Sens. Space Sci.* **2018**, *21*, 363–374. [[CrossRef](#)]
23. Chen, Z.; Liu, C.; Wu, H. A higher-order tensor voting-based approach for road junction detection and delineation from airborne LiDAR data. *ISPRS J. Photogramm.* **2019**, *150*, 91–114. [[CrossRef](#)]
24. Zhu, P.; Lu, Z.; Chen, X.; Honda, K.; Eiumnoh, A. Extraction of city roads through shadow path reconstruction using laser data. *Photogramm. Eng. Remote Sens.* **2004**, *70*, 1433–1440. [[CrossRef](#)]
25. Hu, X.; Tao, C.V.; Hu, Y. Automatic road extraction from dense urban area by integrated processing of high resolution imagery and LiDAR data. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2004**, *35*, 288–292.
26. Youn, J.; Bethel, J.S.; Mikhail, E., M. Extracting urban road networks from high-resolution true orthoimage and LiDAR. *Photogramm. Eng. Remote Sens.* **2008**, *74*, 227–237. [[CrossRef](#)]
27. Wang, G.; Zhang, Y.; Li, J.; Song, P. 3D road information extraction from LiDAR data fused with aerial-images. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Fuzhou, China, 29 June–1 July 2011; pp. 362–366. [[CrossRef](#)]
28. Sameen, M.I.; Pradhan, B. A two-stage optimization strategy for fuzzy object-based analysis using airborne LiDAR and high-resolution orthophotos for urban road extraction. *J. Sens.* **2017**, *2017*, 1–17. [[CrossRef](#)]
29. Milan, A. An integrated framework for road detection in dense urban area from high-resolution satellite imagery and LiDAR data. *J. Geogr. Inf. Syst.* **2018**, *10*, 175–192. [[CrossRef](#)]
30. Nahhas, F.H.; Shafri, H.Z.M.; Sameen, M., I.; Pradhan, B.; Mansor, S. Deep learning approach for building detection using LiDAR–orthoimage fusion. *J. Sens.* **2018**, *2018*, 1–13. [[CrossRef](#)]
31. Maturana, D.; Scherer, S. VoxNet: A 3D convolutional neural network for real-time object recognition. In Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [[CrossRef](#)]
32. Sun, Y.; Zhang, X.; Xin, Q.; Huang, J. Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and LiDAR data. *ISPRS J. Photogramm.* **2018**, *143*, 3–14. [[CrossRef](#)]
33. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [[CrossRef](#)]
34. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1–10.
35. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. Pointsift: A siftlike network module for 3D point cloud semantic segmentation. *arXiv* **2018**, arXiv:1807.00652.
36. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. PointWeb: Enhancing local neighborhood features for point cloud processing. In Proceedings of the IEEE conference on computer vision and pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5565–5573. [[CrossRef](#)]
37. Liu, H.; Guo, Y.; Ma, Y.; Lei, Y.; Wen, G. Semantic context encoding for accurate 3D point cloud segmentation. *IEEE Trans. Multimedia* **2021**, *23*, 2054–2055. [[CrossRef](#)]
38. Feng, M.; Zhang, L.; Lin, X.; Gilani, S.Z.; Mian, A. Point attention network for semantic segmentation of 3D point clouds. *Pattern Recognit.* **2020**, *107*, 107446. [[CrossRef](#)]
39. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE conference on computer vision and pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11105–11114. [[CrossRef](#)]
40. Özdemir, E.; Remondino, F.; Golkar, A. Aerial point cloud classification with deep learning and machine learning algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 843–849. [[CrossRef](#)]
41. Wen, C.; Li, X.; Yao, X.; Peng, L.; Chi, T. Airborne LiDAR point cloud classification with global-local graph attention convolution neural network. *ISPRS J. Photogramm.* **2021**, *173*, 181–194. [[CrossRef](#)]
42. Hu, X.; Li, Y.; Shan, J.; Zhang, J.; Zhang, Y. Road centerline extraction in complex urban scenes from LiDAR data based on multiple features. *IEEE Transact. Geosci. Remote Sens.* **2014**, *52*, 7448–7456. [[CrossRef](#)]
43. Huang, R.; Xu, Y.; Hong, D.; Yao, W.; Ghamisi, P.; Stilla, U. Deep point embedding for urban classification using ALS point clouds: A new perspective from local to global. *ISPRS J. Photogramm.* **2020**, *163*, 62–81. [[CrossRef](#)]
44. Jing, Z.; Guan, H.; Zhao, P.; Li, D.; Yu, Y.; Zang, Y.; Wang, Y.; Wang, H.; Li, J. Multispectral LiDAR point cloud classification using SE-PointNet++. *Remote Sens.* **2021**, *13*, 2516. [[CrossRef](#)]
45. Landrieu, L.; Raguét, H.; Vallet, B.; Mallet, C.; Weinmann, M. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS J. Photogramm.* **2017**, *132*, 102–118. [[CrossRef](#)]
46. Zhang, J.; Lin, X.; Ning, X. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775. [[CrossRef](#)]
47. Lee, J.; Dycziej-Edlinger, R. Automatic mesh generation using a modified Delaunay tessellation. *IEEE Antennas Propagat. Magazine* **1997**, *39*, 3445. [[CrossRef](#)]

48. Kwak, E.; Habib, A. Automatic representation and reconstruction of DBM from LiDAR data using recursive minimum bounding rectangle. *ISPRS J. Photogramm.* **2014**, *93*, 171–191. [[CrossRef](#)]
49. Heipke, C.; Mayer, H.; Wiedemann, C.; Jamet, O. Evaluation of automatic road extraction. *Int. Arch. Photogramm. Remote Sens.* **1997**, *32*, 172–187.
50. Boyko, A.; Funkhouser, T. Extracting roads from dense point clouds in large scale urban environment. *ISPRS J. Photogramm.* **2011**, *66*, S2–S12. [[CrossRef](#)]
51. Shi, X.; Ma, H.; Chen, Y.; Zhang, L.; Zhou, W. A parameter-free progressive TIN densification filtering algorithm for LiDAR point clouds. *Int. J. Remote Sens.* **2018**, *39*, 6969–6982. [[CrossRef](#)]