

Article

# A Multi-Constraint Combined Method for Ground Surface Point Filtering from Mobile LiDAR Point Clouds

Li Yan, Hua Liu \* , Junxiang Tan , Zan Li and Changjun Chen

School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; liyan@sgg.whu.edu.cn (L.Y.); tanjunxiang@whu.edu.cn (J.T.); lizan@whu.edu.cn (Z.L.); chencj@whu.edu.cn (C.C.)

\* Correspondence: liuhua@whu.edu.cn; Tel.: +86-135-5453-5370

Received: 12 August 2017; Accepted: 13 September 2017; Published: 15 September 2017

**Abstract:** Point cloud filtering is an essential preprocessing step in 3D (three-dimensional) LiDAR (light detection and ranging) point cloud processing. The filtering of mobile LiDAR scanning point clouds is much more challenging due to their non-uniform distribution, the large-scale of missing data areas and the existence of both large size objects and small land features. This paper proposes a new filtering method that combines range constraint, slope constraint and angular position constraint to filter ground surface points from mobile LiDAR point clouds. Firstly, a cylindrical coordinate system (CCS) is established for each block of mobile LiDAR point clouds and three attributes of mobile LiDAR points, i.e., the angular position attribute (AA), longitudinal distance attribute (LA) and range attribute (RA), are computed. Then, the mobile LiDAR point clouds are structured into a grid according to the AA and LA. Finally, the point clouds are filtered by a single cross-section filter (SCSF) using range constraint and slope constraint, followed by a multiple cross-section filter (MCSF) using range constraint and angular position constraint. Five datasets are used to validate the proposed method. The experimental results show that the proposed new filtering method achieves an average type I error, type II error, and total error of 1.426%, 1.885%, and 1.622%, respectively.

**Keywords:** mobile LiDAR scanning; point clouds; ground surface points; filtering; classification

## 1. Introduction

Mobile LiDAR scanning (MLS) is a state-of-the-art form of three-dimensional (3D) spatial data acquisition technology and has been widely used in road surface mapping [1], tunnel mapping [2], and road inventory [3], etc. The usage of MLS data can be mainly divided into two categories: topographical mapping of the road corridor and road object information extraction. In topographical mapping of the road corridor, ground surface points should be extracted from the raw MLS data. In road object information extraction, ground surface points are first obtained (for road marking extraction) or removed (for traffic sign and pole-like object extraction) to reduce the data volume and facilitate object extraction. In both categories of MLS data usage, ground surface point filtering plays an important role in data processing.

Many filtering algorithms have been proposed to cope with the ground surface point filtering problem of 3D LiDAR point clouds. The existing filtering methods can mainly be categorized into four groups: (1) slope-based filtering methods; (2) morphology-based filtering methods; (3) surface-based filtering methods; and (4) segmentation-based filtering methods. Vosselman [4] first proposed the slope-based filtering method, in which a point is classified as a ground point if all of the points within a distance threshold have a height difference smaller than the allowed maximum difference. Instead of using the same filter kernel for an entire dataset, Sithole [5] modified the slope-based filtering method in such a manner that the threshold varied with respect to the slope of the terrain and better

performance was achieved in steep areas. Local slope in [4] was refined by the local linear regression criterion in [6]. The minimum height was first found in the neighborhood and points were assigned to the ground if their heights were compatible with the linear regression criterion. Then the approximate DEM (digital elevation model) was calculated according to the weighted mean of the ground points. Finally, the points with distances to the approximate DEM that met the threshold criterion were classified as final ground points.

The morphology-based filtering methods conduct morphological operations on the point clouds and use the height difference between points before and after morphological operations to judge whether the point is a ground point or not. The morphological operator “opening” was applied several times with different window sizes to filter ground points from airborne LiDAR scanning (ALS) data by Kilian et al. [7]. A window was moved over the dataset and weights depending on the window size were assigned to points higher than the deepest point in the applied window. The ground surface was interpolated under the consideration of the weights. Zhang et al. [8] developed a progressive morphological filter to remove non-ground points from ALS data. By gradually increasing the window size, non-ground points were removed progressively. Chen et al. [9] developed a similar progressive morphological filter but the slope was no longer assumed as constant. Pingel et al. [10] implemented a morphological filter with a linearly-increasing window and simple slope thresholding. Mongus et al. [11] filtered ground surface points from ALS data with features mapped from differential morphological profiles.

There are two types of surface-based filtering methods: surface interpolation methods and progressive TIN (triangulated irregular network) densification (PTD) methods. Kraus and Pfeifer [12] and Pfeifer et al. [13] used the full ALS data with equal weights to interpolate an initial surface, and the weight of each point was then updated based on the distance to the interpolated surface. The surface was iteratively interpolated until the weights of all points remained unchanged. Points within the buffer of the final interpolated surface were classified as ground points. An active shape model that acts like a rubber cloth with elasticity and rigidity was used to model the ground surface in [14,15]. The model was fitted by minimizing the energy function and any points within the buffer of the fitted model were accepted as ground surface points. Similar to the active shape model method, cloth simulation was used to interpolate the ground surface in [16]. Hu et al. [17] interpolated thin plate spline surfaces toward the ground in their adaptive surface filter (ASF). Bending energy was embedded in ASF to self-adapt the filter threshold automatically. Axelsson [18] proposed a PTD method to separate ground surface points from non-ground surface points. The seed ground points were used to create an initial TIN structure, and the points meeting distance and angle criteria were added to densify the TIN progressively. A two-step progressive TIN densification method was proposed by Sohn and Dowman [19]. In the downward densification step, four corner points were first chosen and triangulated, then the lowest points in each triangle were added to the triangulation. The downward densification process was repeated until no more points were found beneath each triangle. In the upward densification step, a buffer was defined for each triangle. The point within the buffer that forms the flattest tetrahedral was added to the triangulation. The final TIN represents the ground surface. Zhang and Lin [20] improved the traditional PTD filter by adding segmentation using smoothness constraint (SUSC) between TIN construction and densification to preserve ground points in areas with steep terrain. Zhao et al. [21] improved the traditional PTD filter by combining it with a morphological method to provide more ground seed points, adding simulated ground points to improve the quality of the initial TIN and performing downward densification before upward densification to deal with slope variations.

The segmentation-based filtering methods classify a segment rather than a single point as ground or non-ground based on local geometrical relations such as height, slope or curvature in the neighborhood. Brovelli et al. [22,23] proposed a method to filter ALS data according to the edges detected in the interpolated DSM (digital surface model). The points inside the convex hull determined by the edge cells were classified as objects in case their height was greater than or equal to the computed mean edge height. Tóvári and Pfeifer [24] extended the point-based filtering method [12] to

a segmentation-based method to achieve better performance. The raw ALS data were first segmented into segments by region growing, and then a surface interpolation method similar to [12] was used to interpolate the ground surface.

The filtering algorithms mentioned above were mainly developed for ALS data. Compared with ALS data, MLS data have a much higher point density, which usually greatly decreases the efficiency of filtering algorithms. Besides, the point density of MLS data is not uniform in the data extent because of the different scanning distances, making it confusing to determine the window size and grid size in morphology-based and surface-based filtering algorithms, respectively. In some works [25,26], the authors attempted to calculate the MLS point density accurately to minimize the influence of the non-uniform distribution on MLS data processing. Last but not least, large-scale missing data areas exist inside the MLS data extent because of occlusion. The interpolation of large-scale missing data areas decreases the reliability of data, especially when ground points are mixed with points from vegetation and buildings [4]. Filtering algorithms for MLS data considering the data characteristics are needed.

However, fewer filtering methods are designed specifically for MLS data, taking into consideration the data characteristics. Wu et al. [27] proposed a method to extract ground surface points from MLS data by organizing data into vertical profiles. Candidate ground points were extracted from the vertical profiles by an alpha shape algorithm and ground points were refined by an elevation variance filter. Nurunnabi et al. [28] proposed a locally-weighted regression technique to fit the ground surface iteratively. Points within the specific height buffer of the final fitted ground surface were identified as ground surface points. Zhou et al. [29] proposed a scanline-based segmentation method to filter ground surface points from an ordered MLS point cloud.

In contrast to intentionally filtering ground surface points, in some works the ground points are extracted or removed first in order to identify road object information from MLS data. In these works, the points on the pavement are extracted or eliminated, leaving the ground points outside the pavement unhandled. For example, Guan et al. [30] proposed a method to extract road surface points using curbstone information on both sides of the road. Zheng et al. [31] removed the points on or around the ground by a piecewise elevation histogram segmentation method. These methods are mainly designed to extract or remove the points inside the road pavement, and they are not suitable for applications requiring full ground surface points within the data extent.

Besides filtering ground surface points from MLS data, in some works, the authors try to label the point clouds or classify the whole point clouds into different categories directly. Usually, ground surface points represent one of the categories. Yang and Dong [32] proposed a shape-based method to segment the MLS point clouds of urban scenes into different objects. Luo et al. [33] proposed a patch-based semantic labeling framework to label each point of colorized mobile LiDAR point clouds. Both semantic labeling and direct classification are much more challenging and time-consuming due to the scene complexity and the huge volume of MLS point clouds.

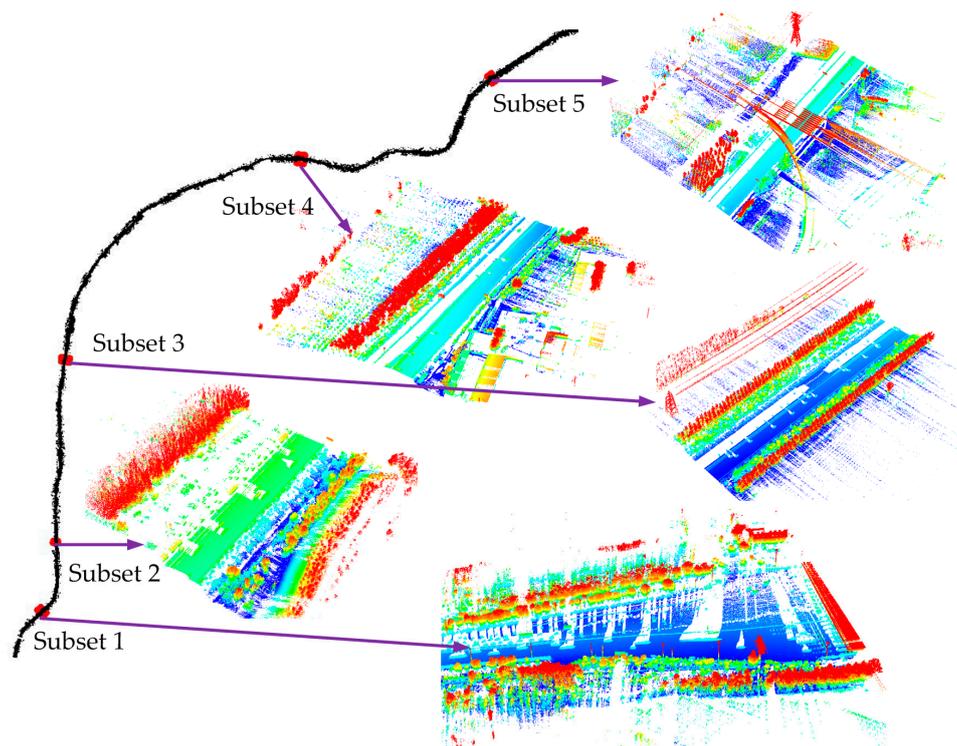
This paper proposes a new filtering method specifically designed to filter ground surface points from MLS data. In our method, a cylindrical coordinate system is established for each block of the mobile LiDAR point cloud and three attributes of mobile LiDAR point clouds, i.e., angular position attribute (AA), longitudinal distance attribute (LA) and range attribute (RA), are first computed. Then, the MLS point clouds are structured into a grid according to AA and LA. The MLS point clouds are quasi-uniformly distributed and large-scale missing data areas are avoided in such a grid. Unlike the existing filtering methods that use slope differences or elevation differences between ground surface points and non-ground surface points, the single cross-section filter (SCSF) using range constraint and slope constraint, and the multiple cross-sections filter (MCSF) using range constraint and angular position constraint are designed to filter ground surface points from MLS data.

The remainder of this paper is organized as follows. The datasets used to validate the proposed filtering method are first described in Section 2, followed by the detailed description of the proposed filtering method. Section 3 presents the experimental results, and the proposed filtering method is discussed in Section 4. Finally, Section 5 concludes this paper.

## 2. Materials and Methods

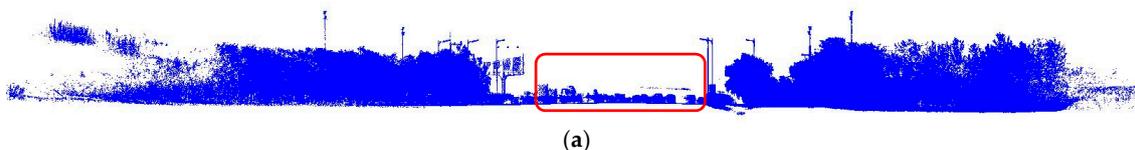
### 2.1. MLS Datasets

The mobile LiDAR point cloud data for the Jincheng highway in Beijing, China were collected by a mobile LiDAR system equipped with a RIEGL VUX-1 laser scanner. The speed of the mobile LiDAR system during data acquisition was about 40 km/h. The angular step width of laser scanning was  $0.02^\circ$  and the distance between two scan lines was about 0.06 m. The total data coverage was about 57 km (Figure 1). For convenience, five subsets with different ground surface types and different non-ground objects were selected as the test data from the whole dataset (Figure 1).

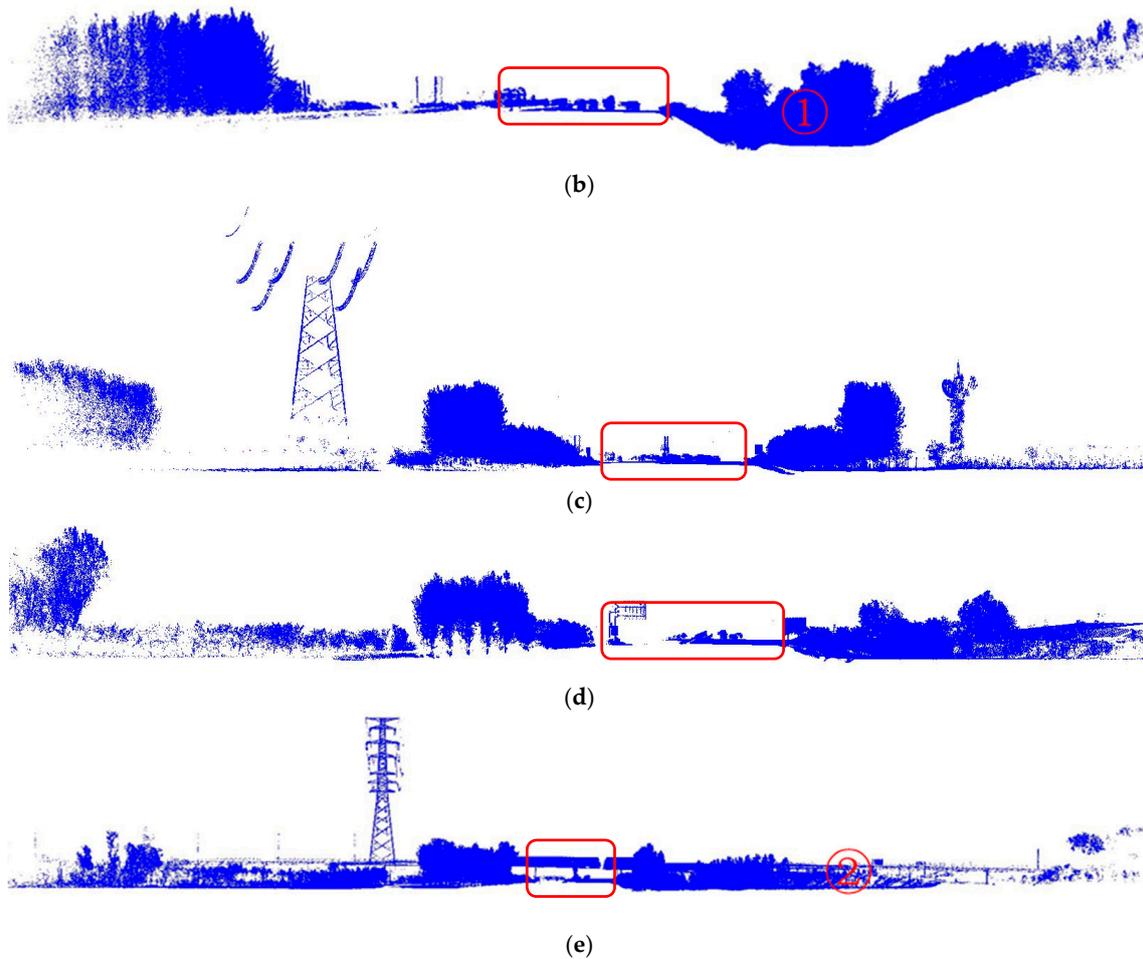


**Figure 1.** The collected mobile LiDAR (light detection and ranging) scanning (MLS) point clouds and five test subsets. The point clouds of the five subsets are colored by elevation.

Different ground surface types and non-ground objects were contained in the selected test data. Figure 2 illustrates the cross-section views of the five datasets. The selected five subsets all have flat road pavements with moving cars on them (the pavements lie inside the red boxes in Figure 2). Besides a flat road pavement, Subset 1, Subset 3 and Subset 4 contain gently sloping ground surfaces, while Subset 2 and Subset 5 contain steep ground surfaces (① in Figure 2b and ② in Figure 2e, respectively). All five subsets contain trees and cars, especially Subset 1 and Subset 2. Lamps exist in Subset 1. Buildings exist in Subset 1 and Subset 4. Power lines exist in Subset 2, Subset 3 and Subset 5. The details of the five selected subsets of data are described in Table 1.



**Figure 2.** Cont.



**Figure 2.** Cross-section views of the five datasets: (a) Subset 1; (b) Subset 2; (c) Subset 3; (d) Subset 4; (e) Subset 5. Roads are inside the red boxes in (a–e).

**Table 1.** The details of the five subset datasets.

Dataset	Length (m)	Number of Points (million)	Ground Surface Type	Typical Non-Ground Objects
Subset 1	295	14.2	Flat Gentle	Large numbers of trees Large numbers of cars Lamps, buildings
Subset 2	140	4.7	Flat Steep	Large numbers of trees Large numbers of cars Power lines
Subset 3	290	9.3	Flat Gentle	Large number of trees Cars, power lines
Subset 4	270	9.6	Flat Gentle	Large numbers of trees Cars, buildings
Subset 5	320	9.6	Flat Steep	Trees, power lines Cars, overpass road

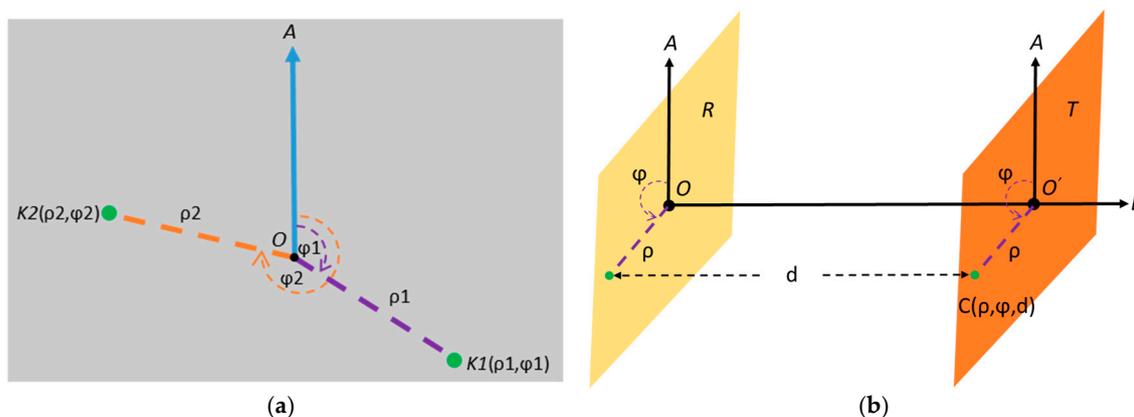
Besides  $x$ ,  $y$ , and  $z$  coordinates, a precise GPS (global positioning system) timestamp is another attribute of each MLS point. The trajectory data, including a precise GPS timestamp and the position and attitude of the mobile LiDAR system platform at that time point, are delivered together with the MLS point clouds.

## 2.2. Computation of MLS Point Cloud Attributes

In order to compute the attributes of the MLS point clouds, i.e., AA, LA and RA, the cylindrical coordinate system was used. AA, LA and RA correspond to the angular position, longitudinal distance and radial distance in cylindrical coordinate system, respectively.

The cylindrical coordinate system can be seen as an extension of a two-dimensional (2D) polar coordinate system. In a polar coordinate system, a point on a plane is determined by a distance from a reference point (origin) and an angle from a reference direction [34]. The distance from the reference point is called the radial distance, and the angle from the reference direction is called the angular position (angular coordinate, or azimuth). Figure 3a is an illustration of a polar coordinate system. In Figure 3a,  $A$  is the reference direction and  $O$  is the reference point (origin). The coordinates of point  $K1$  in the polar coordinate system is determined by the radial distance from  $O$  to  $K1$  ( $\rho_1$ ) and the angle from  $OA$  to  $OK1$  ( $\varphi_1$ ). The coordinates of point  $K2$  are determined by the distance from  $O$  to  $K2$  ( $\rho_2$ ) and the angle from  $OA$  to  $OK2$  ( $\varphi_2$ ).

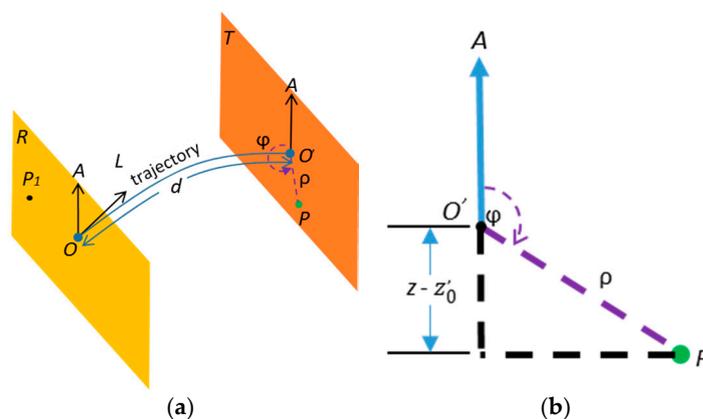
A 3D cylindrical coordinate system combines the 2D polar coordinate system with an additional  $z$ -axis vertically or  $L$ -axis horizontally (this paper uses the  $L$ -axis horizontally) [35]. To construct a 3D cylindrical coordinate system, the reference plane  $R$  should first be obtained. On the reference plane  $R$ , the reference direction  $A$  and the reference point  $O$  are specified to construct a polar coordinate system. The longitudinal axis  $L$  starts from reference point  $O$  and is perpendicular to the reference plane  $R$ , pointing in the positive direction. Figure 3b is an illustration of a 3D cylindrical coordinate system. In Figure 3b,  $T$  is an assistant plane that contains point  $C$  and is parallel to the reference plane  $R$ . The intersection of  $T$  and longitudinal axis  $L$  ( $O'$  in Figure 3b) is the reference point of the polar coordinate system on plane  $T$ , and the reference direction is the same as that in plane  $R$ . The position of point  $C$  in a 3D cylindrical coordinate system can be denoted by the ordered triplet  $(\rho, \varphi, d)$ , in which  $\rho$  and  $\varphi$  are the radial distance and angular position in the polar coordinate system, respectively, while  $d$  is the longitudinal position distance between the assistant plane  $T$  and reference plane  $R$ .



**Figure 3.** Two-dimensional (2D) polar coordinate system (a) and three-dimensional (3D) cylindrical coordinate system (b).

MLS point clouds are usually divided into blocks and the points are ordered by timestamp in each block. To establish a cylindrical coordinate system for each block, the plane that contains the first point  $P1$  (earliest on timestamp) and is perpendicular to the trajectory data is selected as reference plane  $R$ . The intersection of reference plane  $R$  and the trajectory data is defined as the reference point (origin)  $O$ , and the  $z$ -direction is defined as the reference direction  $A$ . In this way, the polar coordinate system in the reference plane is established. The longitudinal axis  $L$  of the 3D cylindrical coordinate system starts at reference point  $O$  and is perpendicular to the reference plane  $R$ , with the direction along the trajectory direction as the positive direction.

Since the trajectory of MLS point cloud data can hardly be a straight line and the longitudinal axis  $L$  will not be parallel to the trajectory data, the conventional cylindrical coordinate system illustrated in Figure 3b should be modified to adapt it to the attributes of the computation of MLS point clouds. First, the assistant plane  $T$  used to figure out the cylindrical coordinates of point  $C$  will no longer be parallel to the reference plane  $R$ . Instead,  $T$  is redefined as the plane that contains point  $C$  and is perpendicular to the trajectory data. The intersection of plane  $T$  and trajectory data is the reference point (origin) of the polar coordinate system on plane  $T$ . The  $z$ -direction is still the reference direction of the polar coordinate system on plane  $T$ . In this way, the radial distance and the angular position of a point in an MLS point clouds can be computed. Second, the longitudinal position  $d$  is not the distance between plane  $T$  and plane  $R$ , but the trajectory length between plane  $R$  and plane  $T$ . Figure 4a is an illustration of the cylindrical coordinate system used to compute the attributes of MLS point clouds.



**Figure 4.** The cylindrical coordinate system used to compute attributes of MLS point clouds (a) and the computation of angular position in the polar coordinate system (b).

As mentioned above, AA, LA and RA correspond to the angular position, longitudinal distance and radial distance in a cylindrical coordinate system, respectively. To compute the attributes of a point  $P$  of MLS point clouds, denoted by  $(t, x, y, z)$ , the Cartesian coordinates of the reference point  $O'$  in plane  $T$ , denoted as  $(x_0, y_0, z_0)$ , are obtained by making  $PO'$  perpendicular to the trajectory. The timestamp  $t$  of point  $P$  is used to find out the trajectory data between  $t - dt$  and  $t + dt$  so as to obtain the reference point in the polar coordinate system more efficiently. Then the RA of  $P$  (denoted as  $\rho$ ) can be obtained by calculating the Euclidean distance between point  $P$  and reference point  $O'$ , according to Equation (1). As illustrated in Figure 4b, the AA of  $P$  (denoted as  $\varphi$ ) can be estimated by the RA and the  $z$  difference between  $P$  and  $O'$  through the trigonometric function as described in Equation (2). The LA of  $P$  is calculated by accumulating the distance from  $O$  to  $O'$  along the trajectory.

$$\rho = \sqrt{(x_0 - x)^2 + (y_0 - y)^2 + (z_0 - z)^2}, \tag{1}$$

$$\varphi = \begin{cases} \arccos\left(\frac{z-z_0}{\rho}\right) & \text{on the right side of trajectory} \\ 2\pi - \arccos\left(\frac{z-z_0}{\rho}\right) & \text{on the left side of trajectory} \end{cases} . \tag{2}$$

### 2.3. Filtering Method

The attributes of MLS point clouds, i.e., AA, LA and RA, are first computed according to Section 2.2. Then, a point in the MLS point clouds is denoted as  $(t, x, y, z, \rho, \varphi, d)$  where  $t, x, y, z, \rho, \varphi, d$  denotes timestamp,  $x$  coordinate,  $y$  coordinate,  $z$  coordinate, range attribute, angular position attribute, and longitudinal distance attribute, respectively. In order to improve the efficiency of the filtering algorithm, a grid is first constructed to establish the connectivity between the MLS points based on

AA and LA. In the constructed grid, a column is a cross-section of the road. To extract the ground surface points, a single cross-section filter (SCSF) using range constraint and slope constraint is first applied to each cross-section. By using SCSF, most of the non-ground points will be filtered out. To refine the ground surface points, multiple cross-sections are used to fit the local surface. The points with an angular position smaller than the minimum angular position or larger than the maximum angular position of that cross-section and the points with a range far away from the fitted local surface are filtered out. The workflow of the filtering method proposed in this paper is illustrated in Figure 5.

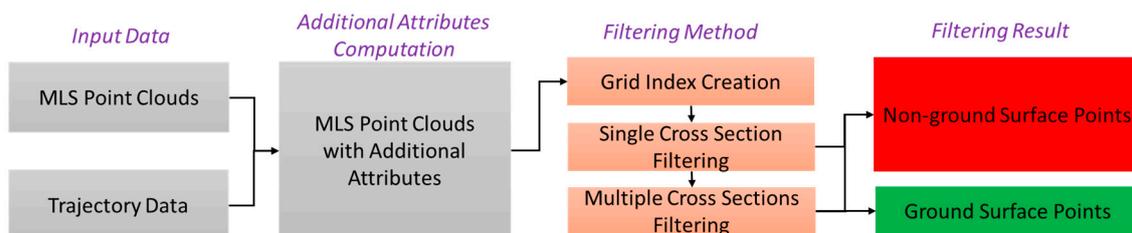


Figure 5. Workflow of the proposed filtering method.

### 2.3.1. Grid Construction

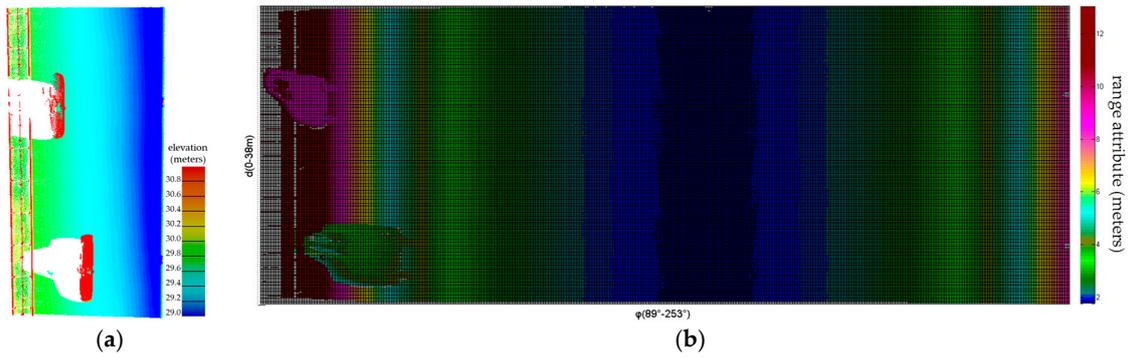
A grid  $g$  is constructed to establish the connectivity between MLS points based on angular position attributes and longitudinal distance attributes. The grid  $g$  is constructed by the following three steps:

- Finding the minimum angular position  $\varphi_{min}$  and the minimum longitudinal distance  $d_{min}$ .
- Determining the angular resolution  $R_a$  and distance resolution  $R_d$ .
- Assigning the MLS point  $P(t, x, y, z, \rho, \varphi, d)$  to the grid cell with row index  $cell_{row}$  and column index  $cell_{col}$  computed using Equations (3) and (4). If more than one point is assigned to the same grid index, the point with largest range attribute is preserved.

$$cell_{row} = \text{Floor}\left[\frac{\varphi - \varphi_{min}}{R_a}\right], \quad (3)$$

$$cell_{col} = \text{Floor}\left[\frac{d - d_{min}}{R_d}\right]. \quad (4)$$

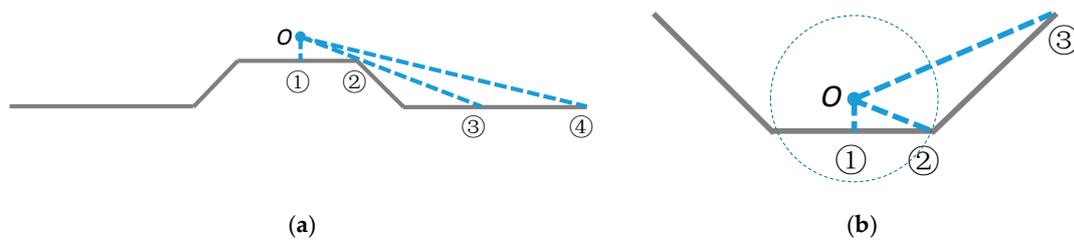
Since the angular step-width of a LiDAR scanner during a project will seldom be changed, the point clouds are quasi-uniformly distributed along the angular position direction even though the angular position is not strictly the same as the scanning angle in laser scanner data acquisition geometry. Besides, as MLS point clouds are usually divided into small blocks and the speed of the platform during a block can be regarded as steady, the MLS points are also quasi-uniformly distributed along the longitudinal distance direction. Compared with the grid created in the XY horizontal plane, the advantage of the grid created in this paper based on the angular position attribute and the longitudinal distance attribute is that the point clouds are quasi-uniformly distributed in the grid and large-scale missing data areas are avoided in such a grid. Figure 6 is an illustration of mobile LiDAR point clouds and the grid created based on the angular position attribute and the longitudinal distance attribute. Figure 6a shows MLS point clouds colored by elevation, while Figure 6b shows the established grid based on the angular position attribute and the longitudinal distance attribute colored by range attribute. In Figure 6b, white cells represent empty cells. It can be seen from Figure 6b that only a few empty cells exist in the grid, which indicates that the point clouds are quasi-uniformly distributed in such a grid and no large-scale empty cells exist in such a grid.



**Figure 6.** MLS point clouds and the corresponding grid created based on the angular position attribute and the longitudinal distance attribute: (a) MLS point clouds colored by elevation (points with low elevation are colored in blue, points with high elevation are colored in red and white areas are those occluded by cars); (b) the grid created based on the angular position attribute and the longitudinal distance attribute colored by range attribute (cells with small range attribute are colored in blue, cells with large range attribute are colored in red and empty cells are colored in white).

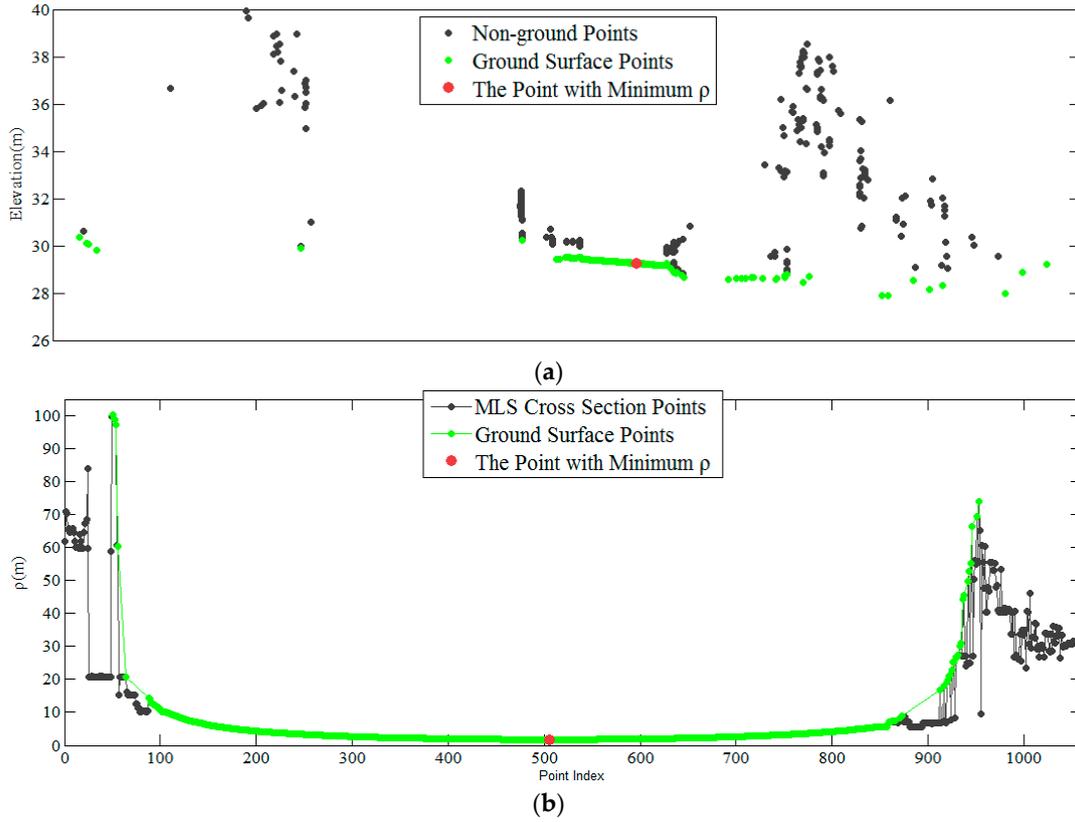
### 2.3.2. Single Cross-Section Filtering (SCSF)

As illustrated in Figure 7, there are two typical types of road cross-section: (1) road cross-sections with a roadside slope lower than the pavement; and (2) road cross-sections with a roadside slope higher than the pavement. *O* in Figure 7a,b represents the origin of the polar coordinate system used to compute the three attributes of MLS point clouds and the blue lines of dashes represent the range attributes of the MLS point clouds. In Figure 7a, the ground surface and objects between ② and ③ are occluded. The range attributes of points from ① to ② and from ③ to ④ will keep increasing obviously. In Figure 7b, the range attributes of points from ① to ② will keep increasing obviously. For points between ② and ③ in Figure 7b, if the next point is outside the circle which centers at *O* with a radius equal to the range attribute of the current point, the range attributes of the points will keep increasing.



**Figure 7.** Two typical types of road cross-section: (a) road cross-section with roadside slope lower than the pavement; (b) road cross-section with roadside slope higher than the pavement.

In the constructed grid *g*, a column is a cross-section of the road. For the points of a cross-section, the range attributes of the ground surface points on both sides of the minimum range point (the point with minimum range attribute in the cross-section) will keep increasing. Figure 8 is an illustration of a cross-section taken from real data. As illustrated in Figure 8b, the range attributes of the ground surface points on both sides of the minimum range point keep increasing. This characteristic can be used to filter ground surface points from MLS point clouds cross-section by cross-section.



**Figure 8.** Characteristics of range attributes in a cross-section: (a) a cross-section of MLS points; (b) the range attributes of each point in the cross-section.

To filter out non-ground surface points in the  $n$ th cross-section, we first found the  $k$ th point  $(t_n^k, \rho_n^k, \varphi_n^k, d_n^k, x_n^k, y_n^k, z_n^k)$  with a minimum range attribute in the cross-section. The filtering is then accomplished by two steps: (1) Searching from  $\varphi_n^k$  to  $\varphi_{Max}(n)$ , the points with  $\rho$  values that continue to increase from  $\rho_n^k$  are classified as ground surface points; and (2) Searching from  $\varphi_n^k$  to  $\varphi_{Min}(n)$ , the points with  $\rho$  values that continue to increase from  $\rho_n^k$  are classified as ground surface points.  $\varphi_{Max}(n)$  and  $\varphi_{Min}(n)$  are the maximum and minimum  $\varphi$  value of the cross-section, respectively. The filtering criteria can be summarized by Equation (5). In Equation (5),  $\max\{\rho_n^{i+1}, \dots, \rho_n^k\}$  and  $\max\{\rho_n^k, \dots, \rho_n^{i-1}\}$  are the maximum  $\rho$  value of the ground surface points within  $(\varphi_n^i, \varphi_n^k]$  and  $[\varphi_n^k, \rho_n^i)$ , respectively.

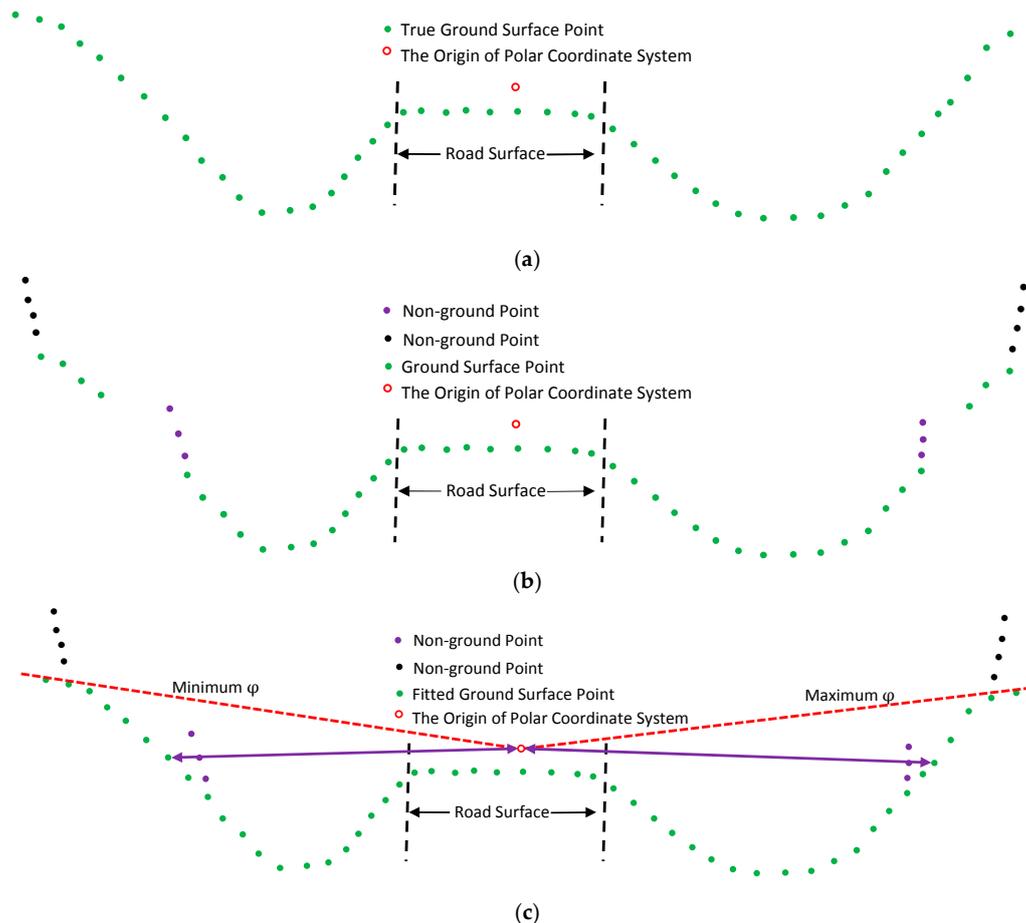
$$\begin{cases} \rho_n^i \geq \max\{\rho_n^{i+1}, \dots, \rho_n^k\} & (\varphi_n^i \leq \varphi_n^k) \\ \rho_n^i \geq \max\{\rho_n^k, \dots, \rho_n^{i-1}\} & (\varphi_n^i > \varphi_n^k) \end{cases} \quad (5)$$

After filtering using the range constraint, the slope constraint is further applied to each cross-section since with the slope constraint it is easy to filter out points on vertical objects and building roofs. The  $k$ th point with minimum  $\rho$  is undoubtedly regarded as a ground surface point. Then the slope constraint is accomplished by two steps: (1) from  $\varphi_n^k$  to  $\varphi_{Max}(n)$ , the slope value of each point  $(t_n^j, \rho_n^j, \varphi_n^j, d_n^j, x_n^j, y_n^j, z_n^j)$  is estimated by Equation (6) and the points with slope values larger than threshold  $T_s$  are classified as non-ground points; and (2) from  $\varphi_n^k$  to  $\varphi_{Min}(n)$ , the slope value of each point  $(t_n^j, \rho_n^j, \varphi_n^j, d_n^j, x_n^j, y_n^j, z_n^j)$  is estimated by Equation (6) and the points with slope values larger than threshold  $T_s$  are classified as non-ground points. In Equation (6),  $x_n^p, y_n^p$  and  $z_n^p$  are the Cartesian coordinates of the last identified ground surface points in  $[\varphi_n^k, \rho_n^j)$  or  $(\varphi_n^j, \varphi_n^k]$ .

$$T_n^j = \text{atan}\left(\frac{\text{abs}(z_n^j - z_n^p)}{\text{sqrt}((x_n^j - x_n^p)^2 + (y_n^j - y_n^p)^2)}\right) \quad (6)$$

### 2.3.3. Multiple Cross-Section Filtering (MCSF)

In SCSF, not all points that satisfy Equation (5) are ground surface points. Figure 9b is an illustration of the result of SCSF. All points in Figure 9b satisfy Equation (5): the green points are ground surface points but the purple and black points are not ground surface points. There are two main types of non-ground points that exist in the result of SCSF: (1) the tree points or vegetation points in the margin of the cross-section (black points in Figure 9b); and (2) the tree points or vegetation points between ground surface points (purple points in Figure 9b).



**Figure 9.** Multiple cross-section filtering: (a) True ground surface points; (b) The result of a single cross-section filter (SCSF); (c) Filtering out the non-ground surface points from the result of the SCSF by the fitted ground surface using multiple cross-sections.

The marginal non-ground surface points in the  $n$ th cross-section could be eliminated by the angular position constraint. If the minimum angular position of ground point  $\varphi_{min}(n)$  and the maximum angular position of ground point  $\varphi_{max}(n)$  of the  $n$ th cross-section are known, the marginal points could be eliminated by eliminating the points with an angular position less than  $\varphi_{min}(n)$  and greater than  $\varphi_{max}(n)$  (the red dash line in Figure 9c). The non-ground surface points between ground surface points could be eliminated by comparing the range attribute with the true surface range in the same grid cell. The range of non-ground surface points is smaller than the true surface range.

The range of the true surface of a cross-section can be fitted by the nearby  $N1$  cross-sections. For the  $i$ th grid cell in the  $n$ th cross-section, the fitted range is the maximum range of the  $i$ th grid cell of all the  $N1$  nearby cross-sections. That means  $\rho_{fn}^i$  can be obtained by Equation (7). Then the fitted range is filtered by SCSF and the range of the filtered-out grid cell is interpolated by the nearby grid cell using linear interpolation. The minimum ground point angular position  $\varphi_{min}(n)$  and the

maximum ground point angular position  $\varphi_{max}(n)$  of the  $n$ th cross-section is estimated by the nearby  $N2$  fitted ground surface cross-sections according to Equations (8) and (9). The non-ground surface points between ground surface points could be eliminated in two steps: (1) for the  $i$ th point between  $\varphi_n^k$  and  $\varphi_{max}(n)$ , if  $\rho_n^i$  is smaller than the range of the  $(i - 1)$ th grid cell in the fitted ground surface, the point is classified as a non-ground point and eliminated; and (2) for the  $i$ th point between  $\varphi_n^k$  and  $\varphi_{min}(n)$ , if  $\rho_n^i$  is smaller than the range of the  $(i + 1)$ th grid cell in the fitted ground surface, the point is classified as a non-ground point and eliminated. The criteria for eliminating non-ground surface points between ground surface points is described in Equation (10).

$$\rho_{f_n}^i = \max\{\rho_{f_{n-\frac{N1}{2}}}^i, \dots, \rho_{f_{n+\frac{N1}{2}}}^i\}, \quad (7)$$

$$\varphi_{min}(n) = \max\left\{\varphi_{min}\left(n - \frac{N2}{2}\right), \dots, \varphi_{min}\left(n + \frac{N2}{2}\right)\right\}, \quad (8)$$

$$\varphi_{max}(n) = \min\left\{\varphi_{max}\left(n - \frac{N2}{2}\right), \dots, \varphi_{max}\left(n + \frac{N2}{2}\right)\right\}, \quad (9)$$

$$\begin{cases} \rho_n^i < \rho_{f_n}^{i+1} (\varphi_n^i \leq \varphi_n^k) \\ \rho_n^i < \rho_{f_n}^{i-1} (\varphi_n^i > \varphi_n^k) \end{cases} . \quad (10)$$

### 3. Results

The five datasets described in Section 2.1 were filtered by the method proposed in this paper. The outlier points with high or low height values were removed manually before filtering. In grid construction, the angular resolution  $R_a$  was set the same as the scanning angular step width ( $0.02^\circ$ ) and the distance resolution  $R_d$  was set the same as the distance between two consecutive scan lines (0.06 m). The slope threshold  $T_s$  was set to  $45^\circ$ . In the MCSF, the number of nearby cross-sections used to interpolate the true ground surface points was set to 50 (about 3 m) and the number of nearby fitted cross-sections used to estimate minimum and maximum angular position of a cross-section was 250 (about 15 m). In order to compare the performance of the proposed filtering method with other filtering algorithms, the five datasets were also filtered by progressive morphological filtering (PMF) [8] and cloth simulation filtering (CSF) [16].

PMF uses a morphological filter to detect and remove non-ground points. The window size of the filter is gradually increased and the elevation difference threshold is determined based on the elevation variations of terrain, buildings and trees. The details of the PMF algorithm can be found in [8]. When filtering with PMF, the maximum window size, the slope value used to compute the height threshold, the initial distance, and the maximum distance were set to 20 m, 0.1, 0.05 m, and 0.5 m, respectively.

In CSF, the point clouds are first inverted and then a rigid cloth is used to cover the inverted surface. The locations of the cloth nodes are determined by analyzing the interactions between the cloth nodes and the corresponding LiDAR point. Ground surface points are obtained by comparing the original points with the generated cloth surface. The details of the CSF algorithm can be found in [16]. When filtering with CSF, the scene types of all the five datasets were set to a steep slope, and post-processing to handle steep slopes was used. The cloth grid resolution and the distance threshold were set to 1.3 and 0.02 m, respectively.

Reference ground surface points for the five datasets were created by the ground surface points filtering method in the commercial software TerraScan and through manual editing. First, the outlier-removed datasets were filtered by TerraScan. When filtering with TerraScan, the maximum building size, the terrain angle, the iteration angle and the iteration distance were set to 15 m,  $45^\circ$ ,  $6^\circ$ , and 0.5 m, respectively. Then the filtering results were manually edited carefully to add ground surface points and remove non-ground surface points. Finally, the points with a distance smaller than 2 cm (a threshold by which almost all pavement points are classified as ground surface points) to the surface constructed by the edited ground surface points were all classified as ground surface points.

The top views and cross-section views of the filtering results created using the proposed method are shown in Figure 10. Since the MLS points in our filtering method are gridded and only one point is kept in a grid, many ground surface points are removed in the grid construction step. In order to evaluate the filtering results more reasonably, the points of filtered ground surface points were used to construct a TIN surface and the original points within 2 cm of the TIN surface were all reclassified as ground surface points. It can be seen from the top views in Figure 10 that the proposed method can correctly classify almost all pavement points as ground surface points and label objects such as cars above the pavement as non-ground points correctly. From the cross-section views in Figure 10 it can be seen that the proposed method can extract most of the ground surface points correctly even in challenging areas such as steep slope areas (Figure 10b2,d2,e2) and under dense vegetation (Figure 10a2,b2,c2). Non-ground points are seldom accepted as ground surface points in such natural landscapes.

To evaluate and compare the performance of the proposed method quantitatively, the type I error, type II error and total error [36] of the filtering results using CSF, PMF, and the proposed method were calculated based on the created reference data and are shown in Table 2. It can be seen from Table 2 that the method proposed in this paper achieves the best type I error in three of the five datasets (Subset 1, Subset 2 and Subset 5), the best type II error in three of the five datasets (Subset 1, Subset 3 and Subset 4) and the best total error in all five datasets. Compared with CSF, the proposed method performs better than CSF in type I error and total error in all five datasets. As for type II error, the proposed method performs better than CSF in gentle slope areas (Subset 1, Subset 3 and Subset 4), while performing slightly poorer in steep slope areas (Subsets 2 and 5). Compared with PMF, the proposed method achieves a comparable type I error in Subsets 1 and 3. The proposed method performs better than PMF for type I error in Subsets 2 and 5, while performing worse in Subset 4. Besides, the proposed method performs better than PMF in type II error, except in Subset 2. The results indicate that the proposed method can achieve good results for all the five datasets, and better results are achieved for ground surfaces with a gentle slope (Subset 1, Subset 3 and Subset 4) than steep ground surfaces (Subsets 2 and 5).

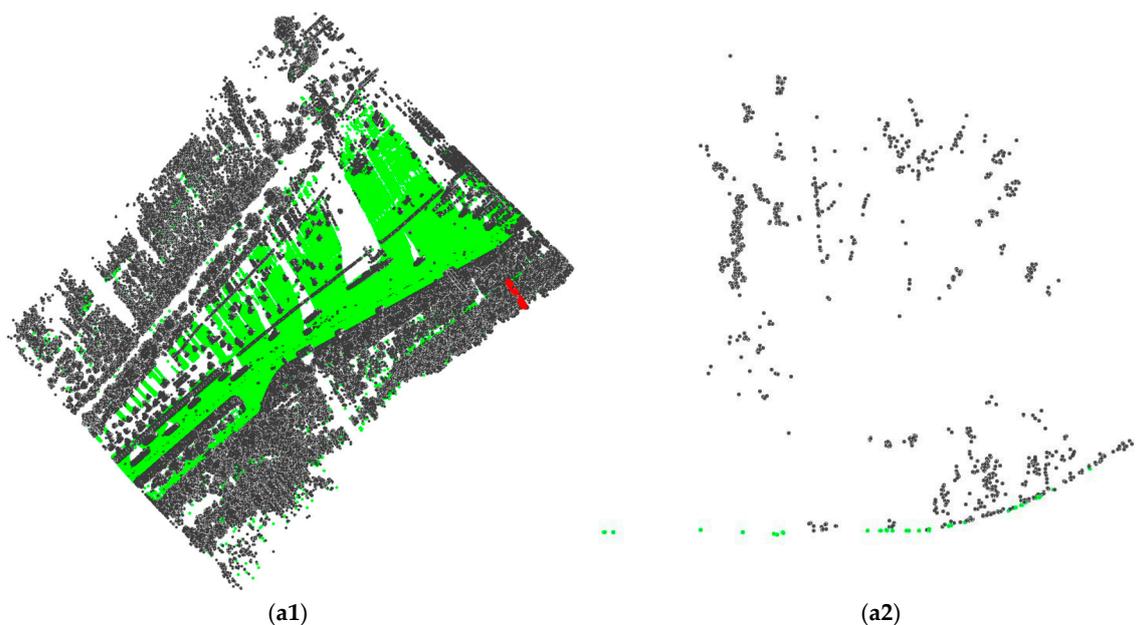
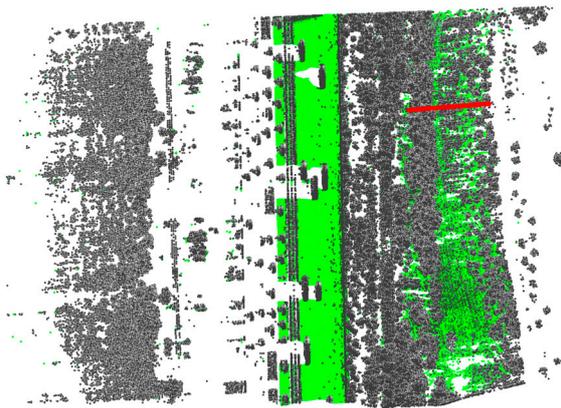
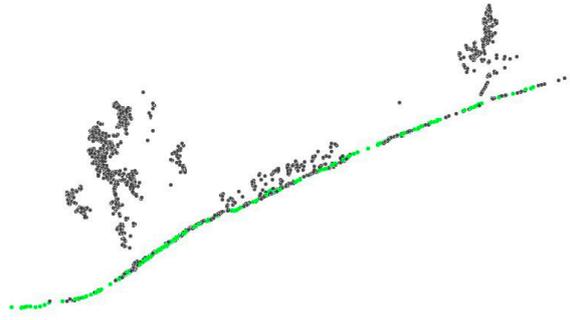


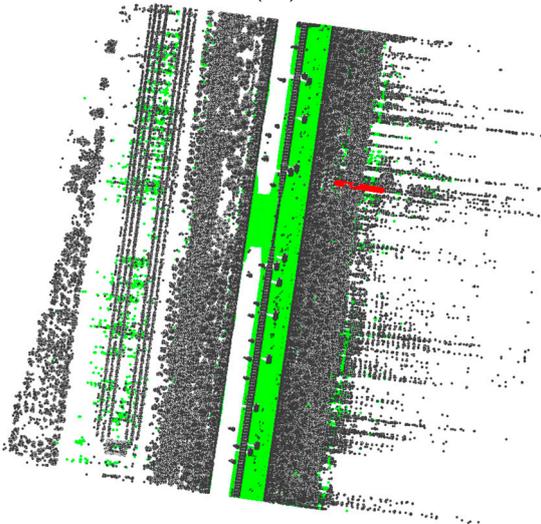
Figure 10. Cont.



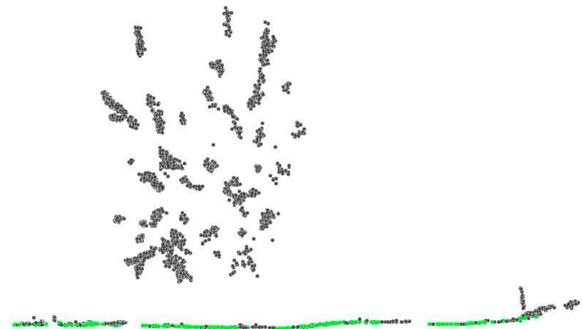
(b1)



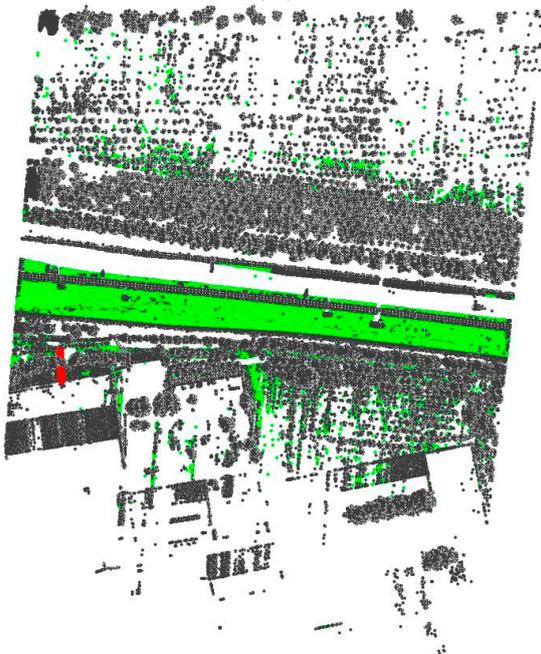
(b2)



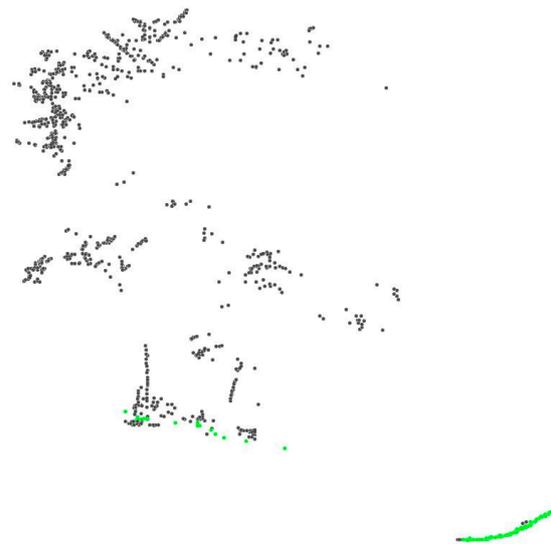
(c1)



(c2)

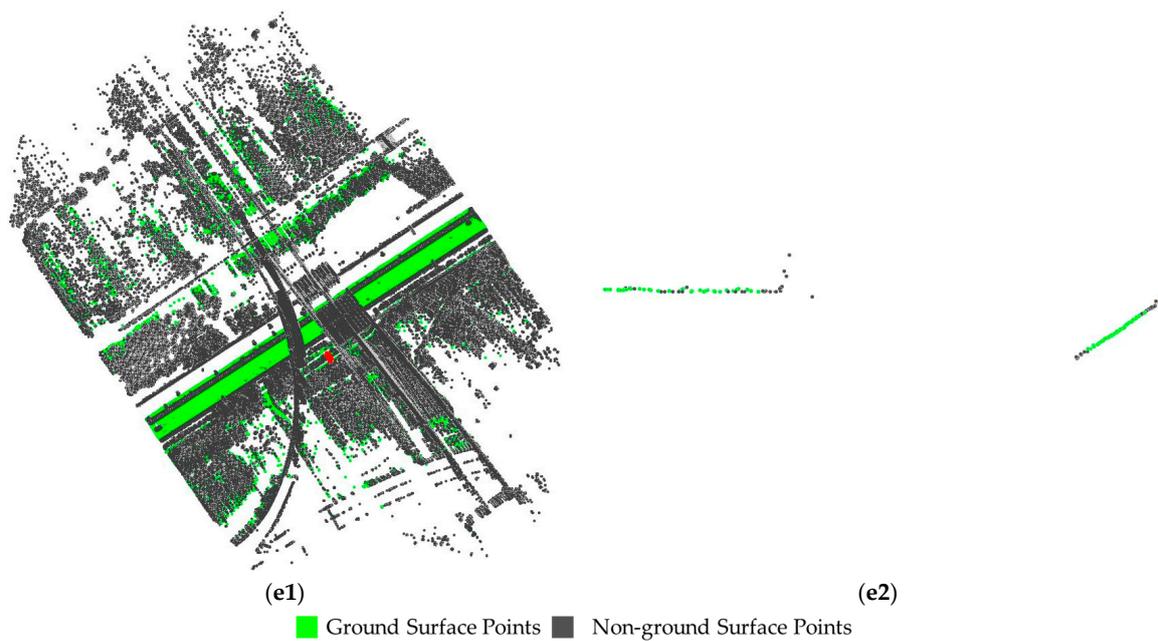


(d1)



(d2)

Figure 10. Cont.



**Figure 10.** Top views (a1,b1,c1,d1,e1) and cross-section views (a2,b2,c2,d2,e2) of filtering results obtained using the proposed method. (a1,b1,c1,d1,e1) are top views of the filtering results of Subset 1, Subset 2, Subset 3, Subset 4 and Subset 5, respectively; (a2,b2,c2,d2,e2) are cross-section views of challenging areas in Subset 1, Subset 2, Subset 3, Subset 4 and Subset 5, respectively. The location of the cross-section views in (a2,b2,c2,d2,e2) are marked by red lines in (a1,b1,c1,d1,e1), respectively.

**Table 2.** Type I error, type II error and total error of filtering results using cloth simulation filtering (CSF), progressive morphological filtering (PMF), and the method proposed in this paper.

Dataset	Type I Error			Type II Error			Total Error		
	CSF	PMF	Proposed Method	CSF	PMF	Proposed Method	CSF	PMF	Proposed Method
Subset 1	7.745%	0.872%	0.790%	1.800%	2.172%	0.830%	5.939%	1.267%	0.802%
Subset 2	12.077%	3.318%	1.872%	1.133%	2.811%	3.741%	7.160%	3.090%	2.712%
Subset 3	3.252%	1.270%	1.470%	2.108%	2.699%	1.382%	2.801%	1.833%	1.435%
Subset 4	1.996%	0.470%	1.253%	2.104%	3.300%	1.266%	2.027%	1.283%	1.256%
Subset 5	14.606%	2.342%	1.743%	1.831%	2.566%	2.204%	10.155%	2.420%	1.904%
Average	7.935%	1.654%	1.426%	1.795%	2.710%	1.885%	5.616%	1.979%	1.622%

The computing time for attribute computation and filtering were recorded to analyze the time complexity of the proposed method. The proposed methods were implemented in C++ and run on an Intel(R) Core(TM) i7-3770 computer. The time costs are shown in Table 3. The results show that more than 80% of the time is spent on attribute computation. Finding the origin of the polar coordinates in attribute computation is time-consuming. However, the time cost is still tolerable, taking the huge amount of point numbers into consideration.

**Table 3.** Time costs of filtering.

Dataset	Attribute Computation (s)	Filtering (s)	Total (s)
Subset 1	1308.5	227.5	1536.0
Subset 2	278.0	30.7	308.7
Subset 3	729.3	95.1	824.4
Subset 4	882.9	106.6	989.5
Subset 5	817.1	145.4	962.5

## 4. Discussion

### 4.1. Accuracy

The proposed filtering method achieved the best total error in all the five datasets and achieves the best type I error and type II error in three of the five datasets. The proposed method achieved inspirational performance even in challenging areas such as steep slope areas and under dense vegetation. The average type I error (1.462%), average type II error (1.885%) and average total error (1.622%) of the proposed filtering method were relatively low compared with PMF and CSF. These results demonstrate that the proposed algorithm can be adapted to the filtering of MLS point clouds in different terrain types and different scene types.

### 4.2. Parameter Setting

The angular resolution  $R_a$  and the distance resolution  $R_d$  in grid construction can easily be set to the same as the scanning angular step width and distance between two consecutive scan lines. In this way, at least one point will be contained in a grid cell in its ideal condition and as few points as possible will be removed due to their falling in the same grid cell.

The number of nearby cross-sections  $N1$  and  $N2$  in MCSF are two parameters that affect the filtering results.  $N2$  is determined by the maximum length of consecutive trees or buildings that prevent the LiDAR signal from hitting the ground surface. It is similar to the parameter maximum building size in the PTD algorithm. In order to filter out the trees and buildings on the margin of both sides of the road, we intended to set a large  $N2$ . It was much harder to choose an optimal  $N1$ , for the true ground surface could not be accurately interpolated with a small  $N1$ , while many ground points will be rejected in steep areas with a large  $N1$  (as illustrated in Figure 11, more ground points are rejected in Figure 11b,c with the increasing  $N1$ ). In 3D point cloud filtering, in order to generate the correct DEM with the filtered ground surface points, a small type II error is preferred, and having some ground surface points rejected is regarded as tolerable. In order to ensure that as few non-ground points as possible exist in the filtering result and the land details are kept as much as possible, we recommend  $N1$  to be set to a value that covers around 3–5 m of the neighborhood.

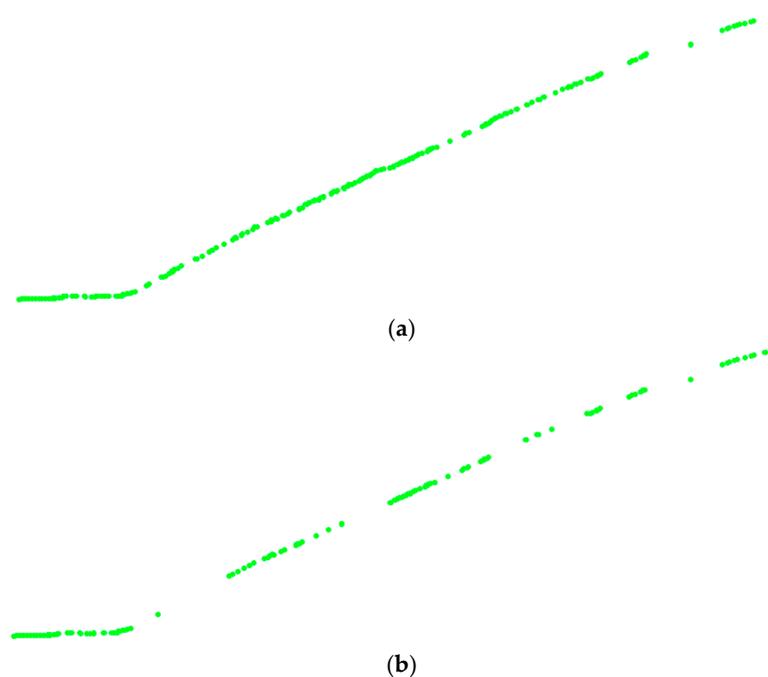
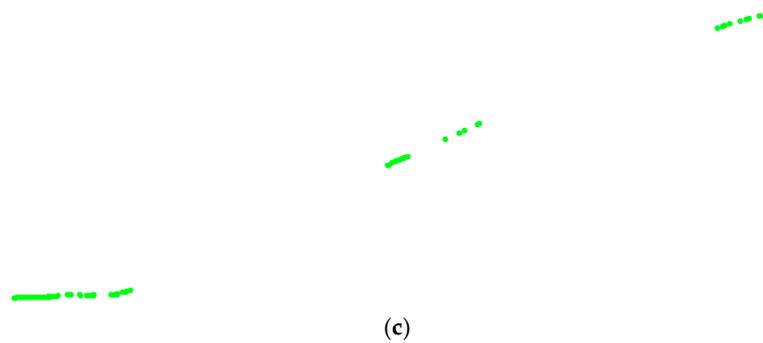


Figure 11. Cont.



**Figure 11.** Filtering results of the steep area with different  $NI$ : (a) filtering result with  $NI$  covers about 0.5 m of the neighborhood; (b) filtering result with  $NI$  covers about 4.0 m of the neighborhood; (c) filtering result with  $NI$  covers about 8.0 m of the neighborhood.

#### 4.3. THE Combination of Multiple Filtering Methods

Even though many algorithms have been proposed, ground surface point filtering from 3D LiDAR point clouds in large-scale complex scenes is still challenging. One of the trends of developing filtering algorithms is to combine multiple filtering methods to achieve better performance in challenging areas. Zhang and Lin [20] combined a segmentation-based method with a progressive TIN-densification method to achieve better performance. Yang et al. [37] combined a segmentation-based filter with a multi-scale morphological filter to remove large objects and preserve landscape details. This paper combines range constraint, slope constraint and angular position constraint to filter ground surface points from MLS point clouds and better performance was achieved compared with PMF and CSF.

## 5. Conclusions

This research proposed a new method to filter ground surface points from MLS point clouds. In contrast to other filtering methods that only use the  $xyz$  Cartesian coordinates, our method first computes attributes, i.e., the angular position attribute, longitudinal distance attribute and range attribute, of MLS point clouds. A grid is then constructed using the angular position attribute and longitudinal distance attribute to establish the connectivity between point clouds. Single cross-section filtering (SCSF) using the range constraint and slope constraint and multiple cross-section filtering (MCSF) using angular position constraint and range constraint are sequentially applied to the constructed grid to get the ground surface points. Five datasets were used to validate the performance of the proposed method. The experimental results show that the average type I error, type II error, and total error of the proposed filtering method in the five datasets were 1.426%, 1.885% and 1.622%, respectively. The average type I error, type II error and total error of the proposed filtering method are relatively low compared with CSF and PMF.

The novelty of the method proposed in this paper can be summarized in two thematic blocks. Firstly, the computed attributes of MLS point clouds. The angular position attribute, longitudinal distance attribute and range attribute provide an alternative for MLS data processing. Besides the SCSF and MCSF proposed in this research, some other algorithms may be more convenient and effective with the help of the angular position attribute, longitudinal distance attribute and range attribute. More algorithms such as pole-like object extraction and tree extraction may be proposed with the help of these attributes. Secondly, the SCSF using the range constraint and slope constraint and MCSF using the angular position constraint and range constraint. The experiments show that the combination of range constraint, slope constraint and angular position constraint can filter ground surface points from MLS point clouds effectively. This also provides an alternative to the combination of multiple filters to achieve better performance in challenging scenes.

Future research should find a more efficient algorithm to compute the attributes, since the computation is time-consuming at this time. Furthermore, the research should move forward to

develop more algorithms such as pole-like object recognition with the help of the computed attributes proposed in this paper.

**Acknowledgments:** This research is supported by the National Natural Science Foundation of China (Grant No. 41771486). The authors would like to thank Xiaoming Zhang of Wuhan University for her valuable suggestions and the anonymous reviewers for their valuable comments.

**Author Contributions:** Hua Liu proposed the original idea; Hua Liu and Junxiang Tan implemented the method; Hua Liu, Junxiang Tan and Zan Li conducted the experiments and analyzed the data; Li Yan and Changjun Chen supervised the whole research.

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

## References

1. Holgado-Barco, A.; Gonzalez-Aguilera, D.; Arias-Sanchez, P.; Martinez-Sanchez, J. An automated approach to vertical road characterisation using mobile LiDAR systems: Longitudinal profiles and cross-sections. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 28–37. [[CrossRef](#)]
2. Puente, I.; Akinci, B.; González-Jorge, H.; Díaz-Vilariño, L.; Arias, P. A semi-automated method for extracting vertical clearance and cross sections in tunnels using mobile LiDAR data. *Tunn. Undergr. Space Technol.* **2016**, *59*, 48–54. [[CrossRef](#)]
3. Guan, H.; Li, J.; Cao, S.; Yu, Y. Use of mobile LiDAR in road information inventory: A review. *Int. J. Image Data Fusion* **2016**, *7*, 219–242. [[CrossRef](#)]
4. Vosselman, G. Slope Based Filtering of Laser Altimetry Data. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 935–942.
5. Sithole, G. Filtering of laser altimetry data using slope adaptive filter. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2001**, *34*, 203–210.
6. Roggero, M. Airborne Laser Scanning-Clustering in Raw Data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2001**, *34*, 227–232.
7. Kilian, J.; Haala, N.; Englich, M. Capture and Evaluation of Airborne Laser Scanner Data. *Int. Arch. Photogramm. Remote Sens.* **1996**, *31*, 383–388.
8. Zhang, K.; Chen, S.C.; Whitman, D.; Shyu, M.L. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 872–882. [[CrossRef](#)]
9. Chen, Q.; Gong, P.; Baldocchi, D.; Xie, G. Filtering Airborne Laser Scanning Data with Morphological Methods. *Photogramm. Eng. Remote Sens.* **2007**, *2*, 175–185. [[CrossRef](#)]
10. Pingel, T.J.; Clarke, K.C.; McBride, W.A. An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *77*, 21–30. [[CrossRef](#)]
11. 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](#)]
12. Kraus, K.; Pfeifer, N. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* **1998**, *53*, 193–203. [[CrossRef](#)]
13. Pfeifer, N.; Reiter, T.; Briese, C.; Rieger, W. Interpolation of high quality ground models from laser scanner data in forested areas. *Int. Arch. Photogramm. Remote Sens.* **1999**, *32*, 31–36.
14. Elmqvist, M. Ground Surface Estimation from Airborne Laser Scanner Data Using Active Shape Models. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 114–118.
15. Elmqvist, M.; Jungert, E.; Lantz, F.; Persson, Å.; Söderman, U. Terrain modelling and analysis using laser scanner data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2001**, *34*, 22–24.
16. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 501. [[CrossRef](#)]
17. Hu, H.; Ding, Y.; Zhu, Q.; Wu, B.; Lin, H.; Du, Z.; Zhang, Y.; Zhang, Y. An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 98–111. [[CrossRef](#)]
18. Axelsson, P. Processing of laser scanner data-algorithms and applications. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 138–147. [[CrossRef](#)]

19. Sohn, G.; Dowman, I. Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 336–344.
20. Zhang, J.; Lin, X. Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 44–59. [[CrossRef](#)]
21. Zhao, X.; Guo, Q.; Su, Y.; Xue, B. Improved progressive TIN densification filtering algorithm for airborne LiDAR data in forested areas. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 79–91. [[CrossRef](#)]
22. Brovelli, M.A.; Cannata, M.; Longoni, U.M. LIDAR Data Filtering and DTM Interpolation Within GRASS. *Trans. GIS* **2004**, *8*, 155–174. [[CrossRef](#)]
23. Brovelli, M.A.; Cannata, M.; Longoni, U.M. Managing and processing LIDAR data within GRASS. In Proceedings of the GRASS Users Conference, Trento, Italy, 11–13 September 2002; Volume 29.
24. Tóvári, D.; Pfeifer, N. Segmentation based robust interpolation—a new approach to laser data filtering. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2005**, *36*, 79–84.
25. Cahalane, C.; McElhinney, C.; Lewis, P.; McCarthy, T. Calculation of Target-Specific Point Distribution for 2D Mobile Laser Scanners. *Sensors* **2014**, *14*, 9471–9488. [[CrossRef](#)] [[PubMed](#)]
26. Cahalane, C.; McElhinney, C.; Lewis, P.; McCarthy, T. MIMIC: An Innovative Methodology for Determining Mobile Laser Scanning System Point Density. *Remote Sens.* **2014**, *6*, 7857–7877.
27. Wu, B.; Yu, B.; Huang, C.; Wu, Q.; Wu, J. Automated extraction of ground surface along urban roads from mobile laser scanning point clouds. *Remote Sens. Lett.* **2016**, *7*, 170–179. [[CrossRef](#)]
28. Nurunnabi, A.; West, G.; Belton, D. Robust Locally Weighted Regression Techniques for Ground Surface Points Filtering in Mobile Laser Scanning Three Dimensional Point Cloud Data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2181–2193. [[CrossRef](#)]
29. Zhou, Y.; Wang, D.; Xie, X.; Ren, Y. A Fast and Accurate Segmentation Method for Ordered LiDAR Point Cloud of Large-Scale Scenes. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1981–1985. [[CrossRef](#)]
30. Guan, H.; Li, J.; Yu, Y.; Wang, C.; Chapman, M.; Yang, B. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 93–107. [[CrossRef](#)]
31. Zheng, H.; Wang, R.; Xu, S. Recognizing Street Lighting Poles From Mobile LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 407–420. [[CrossRef](#)]
32. 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](#)]
33. Luo, H.; Wang, C.; Wen, C.; Cai, Z.; Chen, Z.; Wang, H.; Yu, Y.; Li, J. Patch-Based Semantic Labeling of Road Scene Using Colorized Mobile LiDAR Point Clouds. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1286–1297. [[CrossRef](#)]
34. Stover, C.; Weisstein, E.W. Polar Coordinates. From MathWorld—A Wolfram Web Resource. Available online: <http://mathworld.wolfram.com/PolarCoordinates.html> (accessed on 3 September 2017).
35. Weisstein, E.W. Cylindrical coordinates. In *CRC Concise Encyclopedia of Mathematics*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2002; p. 653.
36. Sithole, G.; Vosselman, G. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2004**, *59*, 85–101. [[CrossRef](#)]
37. Yang, B.; Huang, R.; Dong, Z.; Zang, Y.; Li, J. Two-step adaptive extraction method for ground points and breaklines from lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2016**, *119*, 373–389. [[CrossRef](#)]

