ISPRS Journal of Photogrammetry and Remote Sensing 81 (2013) 44–59

Contents lists available at SciVerse ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification

Jixian Zhang, Xiangguo Lin *

Key Laboratory of Mapping from Space, Chinese Academy of Surveying and Mapping, Beijing 100830, PR China

ARTICLE INFO

Article history:
Received 2 November 2012
Received in revised form 8 April 2013
Accepted 9 April 2013
Available online 9 May 2013

Keywords:
Airborne LiDAR
Filtering
Progressive TIN densification
Point cloud segmentation
Segmentation using smoothness constraint

ABSTRACT

Progressive TIN densification (PTD) is one of the classic methods for filtering airborne LiDAR point clouds. However, it may fail to preserve ground measurements in areas with steep terrain. A method is proposed to improve the PTD using a point cloud segmentation method, namely segmentation using smoothness constraint (SUSC). The classic PTD has two core steps. The first is selecting seed points and constructing the initial TIN. The second is an iterative densification of the TIN. Our main improvement is embedding the SUSC between these two steps. Specifically, after selecting the lowest points in each grid cell as initial ground seed points, SUSC is employed to expand the set of ground seed points as many as possible, as this can identify more ground seed points for the subsequent densification of the TIN-based terrain model. Seven datasets of ISPRS Working Group III/3 are utilized to test our proposed algorithm and the classic PTD. Experimental results suggest that, compared with the PTD, the proposed method is capable of preserving discontinuities of landscapes and reducing the omission errors and total errors by approximately 10% and 6% respectively, which would significantly decrease the cost of the manual operation required for correcting the result in post-processing.

1. Introduction

During the last decade, airborne LiDAR (Light Detection And Ranging), also termed as airborne laser scanning (ALS), has become increasingly popular for various environmental applications, ranging from the reconstruction of digital terrain models (DTMs; Axelsson, 2000), 3D building models (Maas and Vosselman, 1999) and 3D roads (Oude Elberink and Vosselman, 2009) to the detection of individual tree crowns (Koch et al., 2006), measurement of tree height and estimation of other forest stand parameters (Hyyppä et al., 2001). Compared with Interferometric Synthetic Aperture Radar (InSAR) and photogrammetry, ALS has the advantage of recording dense, discrete, detailed and accurate 3D point coverage over both the objects and ground surfaces directly (Shen et al., 2012). Nevertheless, although most of the technical hardware difficulties and system integration problems have been solved, the development of algorithms and methods for interpreting and modeling of LiDAR data is still urgently needed. Similarly to aerial or satellite optical imagery, extensive post-processing is still required to extract accurate terrain or semantic information from the LiDAR point cloud (Zhang, 2010). One of the post-processing methods needed is filtering. In nearly all LiDAR applications, ground filtering is a necessary step to determine which LiDAR returns are from the ground surface and which are from non-ground surface features (Meng et al., 2010). Consequently, various kinds of filtering methods have been proposed for automatically extracting the ground points from the ALS point clouds. There are different classification systems of the existing filtering methods. According to directional scanning (Meng et al., 2010), current methods can be grouped into two major categories: neighborhood-based approaches and directional filtering (Meng et al., 2009). According to the definition of the ground, Shan and Sampath (2005) separated existing filtering methods into two classes, including labeling and adjustment approaches. According to the filter concept, Sithole and Vosselman (2004) separated existing filtering methods into four classes, i.e., slope-based, block-minimum, surface-based, and clustering/segmentation algorithms. The last group is adopted in this paper. An experimental comparison of the performance of eight filtering algorithms was presented by Sithole and Vosselman (2004). They came to the conclusion that the surface-based filters often yield better results concerning the filter strategy, because they use more context than other filter strategies. The basic idea of surface-based methods is to create a parametric surface with a corresponding buffer zone above it, the surface locates the buffer zone, and as before the buffer zone defines a region in 3D space where ground points are expected to reside (Sithole and

* Corresponding author. Tel.: +86 10 63880577; fax: +86 10 63880535.
E-mail address: linxiangguo@gmail.com (X. Lin).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2013.04.001

Vosselman, 2004). Thus, the core step of this kind of methods is to create a surface approximating the bare earth (Chen et al., 2007). Depending on the means of creating the surface, surface-based filtering methods can be further divided into the following three subcategories:

(1) *Morphology-based filters*. This group of filters originates from mathematical morphology (Soille, 2003). The idea of morphological methods is approximating the terrain surface using morphological operations such as opening or geodesic reconstruction. Kilian et al. (1996) used multiple structure elements (windows) in the morphological opening to remove the objects with different sizes. Zhang et al. (2003) described a similar method using different window sizes and considered height differences within each window, which also provides a method for choosing these parameters. However, their method requires the assumption that the terrain slope is constant. Chen et al. (2007) further improved a morphological filter without the constant slope restriction. Arefi and Hahn (2005) presented a geodesic morphological reconstruction-based algorithm to produce the bare ground. The biggest challenge for these methods is how to keep the terrain features unchanged when the window sizes are changed.

(2) *Iterative-interpolation-based filters*. The second group of algorithms is based on a surface model through the entire point set that iteratively approaches the ground surface. A first and rough surface model is used to calculate residuals from this surface model to the points. If the measured points lie above it, they have less influence on the shape of the surface in the next iteration, and vice versa. Kraus and Pfeifer (1998) used weighted linear least squares interpolation to iteratively approximate the ground surfaces. The basic idea of this method is that ground points usually have negative residuals and objects points have positive ones, thus a weight function was designed to assign high weights to the points with negative residuals and low weights to the points with positive residuals. Pfeifer et al. (2001) embedded this method in a hierarchical approach to handle large buildings and decrease computation costs. Chen et al. (2012) also presented a hierarchical recovery method for filtering. The biggest challenge for these methods is how to increase the efficiency when the accuracy is fixed.

(3) *Progressive-densification-based filters*. Similarly to iterative-interpolation-based filters, this third group of filters works also progressively, where more and more points are classified as ground points. However, the difference between them is that there is no need for interpolation for the later group. Axelsson (2000) first divides the whole point dataset into tiles, and selects the lowest points in each tile as the initial ground points, and then a triangular irregular network (TIN) of those ground points is constructed as the reference surface. For each triangle, one of the still unclassified points being inside is added to the set of ground points if both of the following two criteria are met: the point's distance to the TIN facet and the angle between the TIN facet and the line connecting the point with the facet's closest vertex (see Fig. 2b) must not exceed given thresholds. Before continuing the next iteration, all ground points classified in the current iteration step are added to the TIN. In this way, the triangulation is progressively densified until all points are classified as either ground or object. Axelsson's method is known as progressive TIN densification (PTD) (Sithole and Vosselman, 2004), adaptive TIN models (Axelsson, 2000), or TIN densification (Zhang et al., 2003). PTD is adopted herein.

Among the surface-based filtering methods, PTD is widely employed by both the scientific community and engineering applications, because it has been integrated into the commercial software TerraSolid. However, discontinuities in the bare earth also pose great challenges to the surface-based filters. Theoretically, a ridge or hilltop that is locally higher than other portions of the ground surface may resemble an above-ground object and is difficult to retain, as it is regarded as an artificial object (Shao and Chen, 2008). Fig. 1 shows the PTD filter fails to detect the grounds points around the break lines and steep terrain, which increases the omission errors and cause gaps in the natural terrain. Further experience with manual filtering suggests that it is far harder to fix the omission errors (the percentage of bare earth returns misclassified as object returns) than the commission errors (the percentage of object returns misclassified as bare earth returns) in filtering (Sithole and Vosselman, 2004). This raises the question whether the filtering algorithms can be tuned better to reduce the omission errors, even if this is at the expense of an increased number of commission errors.

The objective of this paper is to propose an improved surface-based filtering method that is capable of removing non-ground objects and preserving terrain features even in steep landscapes and mixed landscapes of urban and forested areas. The method is developed based on the fact that, to a large extent, the terrain can be described by smooth surfaces (Vosselman, 2009), and the existing point cloud segmentation algorithm segmentation using smoothness constraint (SUSC; Rabbani et al., 2006) is capable of recognizing smooth surfaces in a ALS point cloud. SUSC is capable of partitioning the input point cloud into mutually disjoint, smoothly connected regions; and it uses a criterion based on a combination of surface normal similarity and spatial connectivity, which is called smoothness constraint. Moreover, SUSC only needs a few input parameters which can be adjusted to get a desired trade-off between under- and over-segmentation. Thus, we will embed the above point cloud segmentation method into the classic PTD filter, which will promote the classic filter's robustness to various types of complex landscapes. The main contribution of this paper is that point cloud segmentation is employed to detect more ground measurements on smooth terrains as seed points, which is helpful to greatly decrease the omission errors in filtering. To evaluate its performance, this method is compared with classic PTD using the benchmark data provided by ISPRS Commission III/WG 3 (Sithole and Vosselman, 2004), which includes seven sites ranging from urban to rural areas with different complexity.

2. Methods

We improve the PTD filter using the SUSC. Details of both underlying algorithms and our proposed method are described as follows.

2.1. Classic PTD method

Despite many filters having been proposed, most of the details of the filtering algorithms have seldom been reported due to the tendency of some commercial and academic practitioners to keep their work proprietary (Chen et al., 2007). Axelsson (2000) introduced the core steps of PTD but without giving more details. Additionally, the manual of TerraSolid (Terrasolid Ltd., 2010) illustrates the needed parameters about PTD but without publication of the source code of the algorithm. Therefore, the classic PTD has been implemented from scratch in this paper, and it is composed of five steps as follows.

(1) Removing outliers

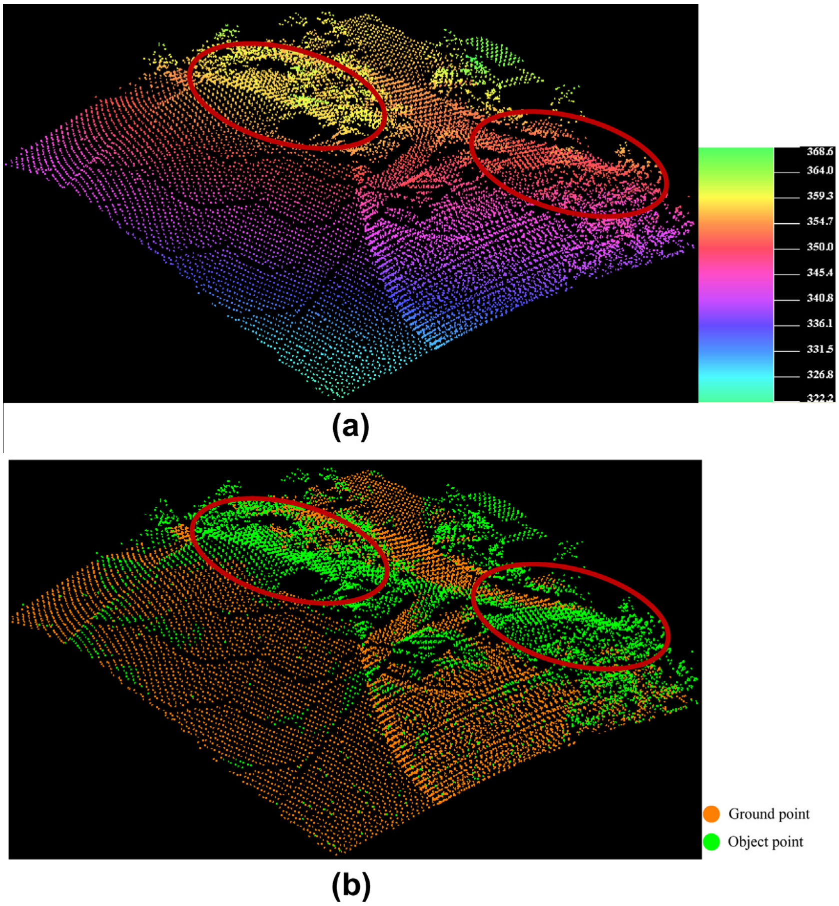


Fig. 1. Most ground measurements around break lines (enclosed by the ellipses) are not detected (they are wrongly classified as "off-terrain") by the classic PTD filter: (a) original point cloud; (b) filtering result of the point cloud in (a).

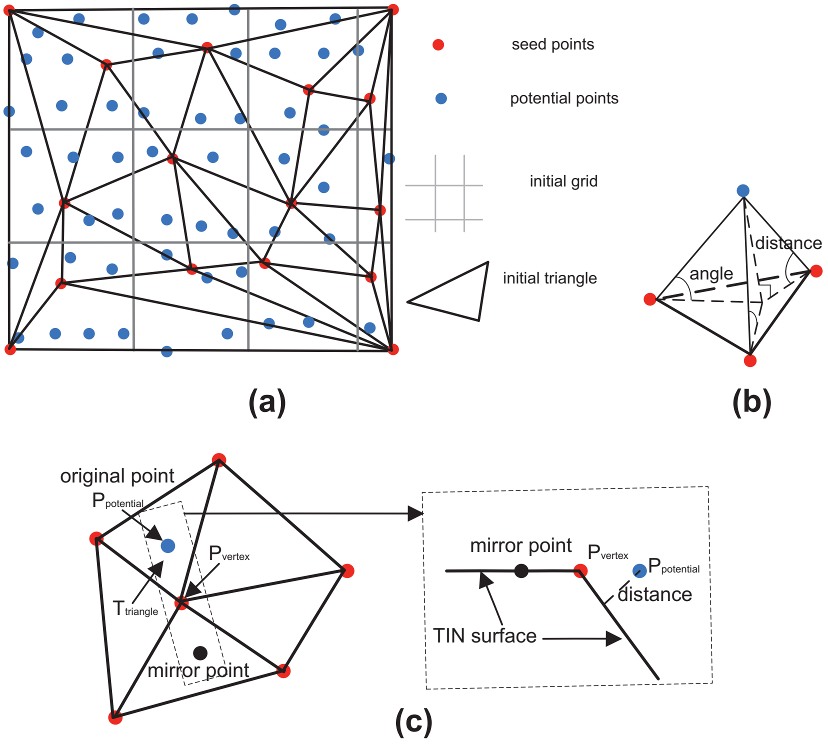


Fig. 2. Parameters in classic PTD: (a) selection of ground seed points and construction of TIN; (b) measurement of angle and distance; (c) mirroring process.

Many datasets contain measurements that are far above or below the earth surface, and these measurements are called outliers. Outliers in the data are one of the circumstances under which the filtering algorithms are likely to fail (Sithole and Vosselman, 2004), especially for the filters with the assumption that the lowest point in a grid cell must belong to the terrain. As a result, three sub-steps are designed to eliminate the outliers. Firstly, the elevation histogram is built and examined by visualizing the distribution of elevation values, and then elevation thresholds were determined to eliminate the lowest and highest tails from the distribution. Secondly, the remaining outliers were searched using the minimum height difference of each point with respect to all its neighbors. Herein, a 2D kd tree is employed to query the neighbors of each point. Points that are too high or too low with respect to their neighbors, are removed from the dataset. Thirdly, errors yielded by the automatic outlier classification are corrected manually.

(2) Specifying parameters

There are six key parameters in PTD to be preset, including:

1) Maximum building size, m. m is a length threshold, and the algorithm can deal with buildings having a length of up to this value. It is used to define the grid cell size.
2) Maximum terrain angle, t. t is a slope threshold, which decides how the judgment of a unclassified point is performed (either mirroring or not). If the slope of a triangle in the TIN is larger than t, any unclassified/potential point located inside of this triangle should be judged by a corresponding mirror point. More details are presented in sub-step 4) of step (4) illustrated in Fig. 2c.
3) Maximum angle, θ. θ is the maximum angle between triangle plane and a line connecting a potential point with the closest triangle vertex. If an unclassified point has a larger angle than θ, it is labeled as an object measurement, otherwise as a ground measurement.
4) Maximum distance, d. d is the maximum distance from a point to triangle plane during one iteration. If an unclassified point has a larger distance than d, it is labeled as an object measurement, otherwise as a ground measurement.
5) Minimum edge length, l. l is the minimum threshold for the maximum (horizontally projected) edge length of any triangle in the TIN. l is utilized to reduce the eagerness to add new points to the ground inside a triangle when every edge of a triangle is shorter than l. Note that l is measured in the horizontal plane. Thus, introduction of l helps to avoid adding unnecessary point density to the ground model and reduces memory requirements.
6) Maximum edge length, l'. l' is the maximum threshold for the minimum (horizontally projected) edge length of any triangle in the TIN. l' is utilized to quit processing a triangle when every edge of the triangle is shorter than l'. Thus, introduction of l' also helps to avoid adding unnecessary point density to the ground model and reduces memory requirements. However, this parameter is employed to thin the ground points, and it has no relationship to filtering. As a result, this parameter is not considered in this paper.

In conclusion, five parameters (m, t, θ, d, and l) need to be preset for the PTD herein.

(3) Selecting seed points and constructing the TIN

Determine the bounding box of the given point cloud dataset, and fix the top left corner ($x_{topleft}, y_{topleft}$), bottom right corner ($x_{bottomright}, y_{bottomright}$), width w and height h. The whole region of dataset is divided into several tiles in rows and columns. Number of rows and columns are determined by the following formula:

$$n_{Row} = ceil\left(\frac{h}{m}\right) \text{ and } n_{Column} = ceil\left(\frac{w}{m}\right) \tag{1}$$

where n_{Row} is the number of tiles in rows, n_{Column} is the number of tiles in columns, $ceil(x)$ is used to return the smallest integral value that is not less than x. The lowest point in each tile is selected as a seed point. Additionally, the four corners on the bounding box should be added to the seed points, as shown in Fig. 2a. Moreover, each corner's height is equal to the one of its closest seed point on horizontal plane. At last, a TIN is constructed based on the seed points, as shown in Fig. 2a, and it represents an initial terrain model. Note the insertion of the four corners guarantees that any point in the point cloud dataset is located inside the TIN. After the TIN is constructed, the remaining points, except the seed points, are labeled as default object measurements.

(4) Iterative densification of the TIN

In each iteration, judging is performed in a point-wise style. That means a potential point is judged based on the input threshold parameters. In detail, the judgment is made as follows:

1) Locate the potential point, $P_{potential}(x_P, y_P, z_P)$. Find the triangle, $T_{triangle}$, which the $P_{potential}$ is inside or on the edge of or on the vertex of.
2) Calculate the slope of the triangle plane, $S_{triangle}$. If $S_{triangle}$ is no larger than terrain angle t, go to step 3). Otherwise, go to step 4).
3) Calculate the following two parameters, as shown in Fig. 2b. This first is the angle between $T_{triangle}$ and a line connecting $P_{potential}$ with the closest triangle vertex, denoted as A_{angle}. The second is the distance from $P_{potential}$ to $T_{triangle}$, denoted as $D_{distance}$. If both of the following cases:

- A_{angle} is no larger than θ,
- $D_{distance}$ is no larger than d,

are held, label $P_{potential}$ as ground measurement. Go to judgment of next point.

4) Mirroring $P_{potential}$, as shown in Fig. 2c. Find the vertex with highest elevation value in $T_{triangle}$, denoted as $P_{vertex}(x_v, y_v, z_v)$. $P_{potential}$ is mirrored as follows:

$$\begin{aligned} x_{mirror} &= 2x_v - x_P \\ y_{mirror} &= 2y_v - y_P \\ z_{mirror} &= z_P \end{aligned} \tag{2}$$

where ($x_{mirror}, y_{mirror}, z_{mirror}$) are the 3D coordinates of the mirror point. Locate the mirror point, and calculate the angle and distance parameters as done in step 3). If the mirror point is determined as a ground point, label $P_{potential}$ as ground measurement, and go to judgment of next point.

At the end of each iteration, the newly detected ground points are added into the TIN as follows:

(1) Locate the ground point, $P_{ground}(x_g, y_g, z_g)$. Find the triangle, $T'_{triange}$, which the P_{ground} is inside or on the edge of or on the vertex of.
(2) Calculate the length of each edge of $T'_{triange}$ in horizontal plane. If the length of any edge is larger than l, add P_{ground} into the TIN and refresh the TIN. Otherwise, go to the judgment of the next newly detected ground point.
(3) Repeat the above iteration until no further point has been added to the set of ground measurements anymore.

The above classic PTD filtering method has been widely applied to various types of landscapes. However, it is still sensitive to steep terrain despite mirroring being adopted, as shown in Fig. 1.

2.2. Point cloud segmentation method

Practice shows that the variety of 3D objects and the massive amount of points, require introduction of some level of organization into the data before the extraction of information can become effective (Filin and Pfeifer, 2006). Such organization is segmentation. Point cloud segmentation is the process to partition an input point cloud into coherent and connected point clusters (Melzer, 2007). Specifically, points on a certain geometric feature are coherent points, such as co-plane, co-surface, and co-line points; whilst connected points are a group of points in which every point has at least one neighboring point within a certain distance (Wang and Tseng, 2010). There are a variety of airborne LiDAR point cloud segmentation methods, such as the surface growing method (Vosselman et al., 2004; Vosselman and Klein, 2010), scan line segmentation method (Sithole, 2005), slope adaptive neighborhood method (Filin and Pfeifer, 2006), and the octree-based split-and-merge method (Wang and Tseng, 2010). Of all these methods, the algorithm about SUSC is selected in this paper. This segmentation method has the following two stages.

(1) Normal and residual estimation

The normal for each point is estimated by fitting a plane to some neighboring points. Therefore, k nearest neighbors (KNNs) (Arya et al., 1998) is employed for the neighborhood search. To fit a plane to a set of given points, in a least squares sense, we need to find the parameters that minimize the sum of squares of the orthogonal distances of the points from the estimated surface. For details of plane fitting refer to Rabbani et al. (2006). The residual in the plane fitting arises from nonconformity of the neighborhood of a point to the planar model, which means that the residual can be used as an approximation of the curvature of the point (Rabbani et al., 2006).

(2) Region growing

This stage makes use of the calculated point normals and their residuals, in accordance with user specified parameters to cluster points belonging to the smooth surfaces. The process of region growing proceeds in the following steps (Rabbani et al., 2006):

(1) Input a residual threshold, r. In this paper, r is calculated from the data using a specified percentile of the sorted residuals in a descending order. A percentile of 95% turned out to be a good choice.
(2) Input a smoothness threshold in terms of the angle between the normals of a seed point and its neighbors, denoted as α.
(3) If all the points have already been segmented, go to step (7). Otherwise select the point with the minimum residual from the set of the still unlabeled points as the current seed, and build an empty list of seed points.
(4) Determine the k nearest neighboring points of the current seed. Add the points, whose angle difference to the current seed is less than α, to the current region; simultaneously, add the points, whose residuals are less than r, to the list of seed points.
(5) If the seed point list is not empty, set the current seed point to the next available seed point from the above list, and go to step (4).
(6) Add the current region to the segmentation and go to step (3), and clear the list of seed points.
(7) Finish the task of segmentation.

The above SUSC algorithm needs an adaptive parameter r and two specified parameters, number of neighbors k, and angle α, these two parameters should be determined based on experience and the complexity of the landscapes. Among the three parameters, r and α define the intended level of smoothness. The result of the SUSC for the point cloud in Fig. 1a is displayed in Fig. 3. Fig. 3 suggests that, after segmentation, the ground surface is grouped into 2 dominant clusters, and the objects are also grouped into many clusters. Moreover, most of the clusters contain a clear majority of either ground or object points, whereas there are hardly any mixed clusters. Particularly, despite the terrain clusters being crossed by breaklines, terrain clusters may contain points on both sides of slope discontinuities.

2.3. Improved method

The analysis in Section 2.1 suggests that the classic PTD is still sensitive to steep terrain and breaklines, and the SUSC method has the capability of grouping the smooth terrains into the same clusters, even complex landscapes. Thus, we will improve the classic PTD using SUSC. The flow chart of the improved method is displayed in Fig. 4.

The main idea of combined PTD/SUSC usage is that all points being located inside the same cluster (obtained by SUSC) as any ground seed point (obtained by the PTD) are most likely ground points as well. Our improved method is composed of six core steps:

(1) Removing outliers, as done in Section 2.1.
(2) Specifying parameters, as done in Section 2.1; additionally, input of two additional parameters of the SUSC, i.e., k and α, is needed.
(3) Selecting seed points and constructing a Delaunay TIN, as done in Section 2.1.
(4) Ground surface growing. This step is modified from the SUSC, with the following two differences. The first one is that each of the initial ground seed points determined by step (3) is selected as seed point one after each other unless it has already been labeled in the improved method, rather than the points with the minimum residuals for the classic SUSC. The second one is that the segmentation would be finished if all initial ground points have been labeled. This improved segmentation process will separate the whole dataset into two subsets, the segmented points with label number and the unsegmented points without label number. As shown in both of Figs. 5d and 6d, many points are labeled by the above point cloud segmentation process, which is meaningful to the subsequent steps. Reclassify the segmented points with label number as ground measurements, and add the newly detected ground points into the TIN and refresh the TIN.
(5) Iterative densification of the TIN, as done in the classic PTD.
(6) Repeat the above iteration until no further point can be added to the set of ground measurements anymore.

3. Experiments and performance evaluation

A prototype software system for filtering ALS data has been developed on a computer with Intel Pentium 2.40 GHz CPU and 2.98 GB RAM using VC++6.0 IDE under the Windows XP Operating System. The PTD method (Axelsson, 2000), SUSC method (Rabbani et al., 2006), and our proposed approach are integrated into the developed system. Additionally, the triangulation of the ALS points was done by integrating an existing implementation of a 2D Delaunay triangulator called Triangle (Shewchuk, 2005), and the KNN

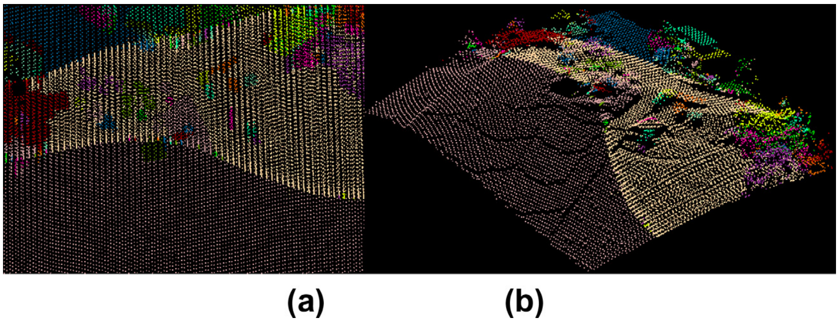


Fig. 3. Result of SUSC with k = 30 and α = 30°: (a) top view of the result; (b) perspective view of the result. Note the point cloud is colored by labeling number in both (a) and (b).

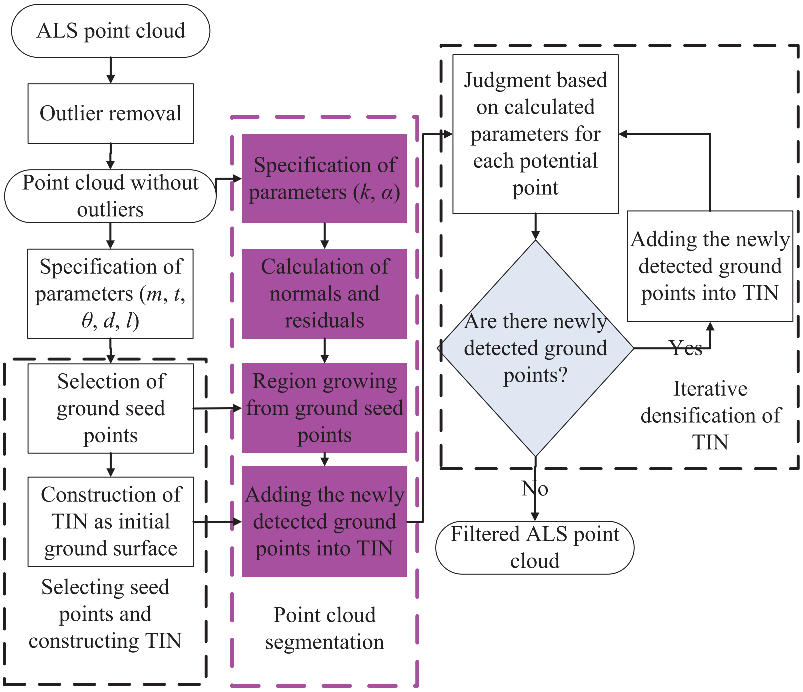

ALS point cloud
Outlier removal
Point cloud without outliers
Specification of parameters (m, t, θ, d, l)
Selection of ground seed points
Construction of TIN as initial ground surface
Selecting seed points and constructing TIN
Specification of parameters (k, α)
Calculation of normals and residuals
Region growing from ground seed points
Adding the newly detected ground points into TIN
Point cloud segmentation
Judgment based on calculated parameters for each potential point
Are there newly detected ground points?
Yes
Adding the newly detected ground points into TIN
Iterative densification of TIN
No
Filtered ALS point cloud

Fig. 4. Flow chart of our improved filtering method.

was done by integrating an existing implementation of kd tree called ANN (Mount and Arya, 2010).

3.1. The testing data

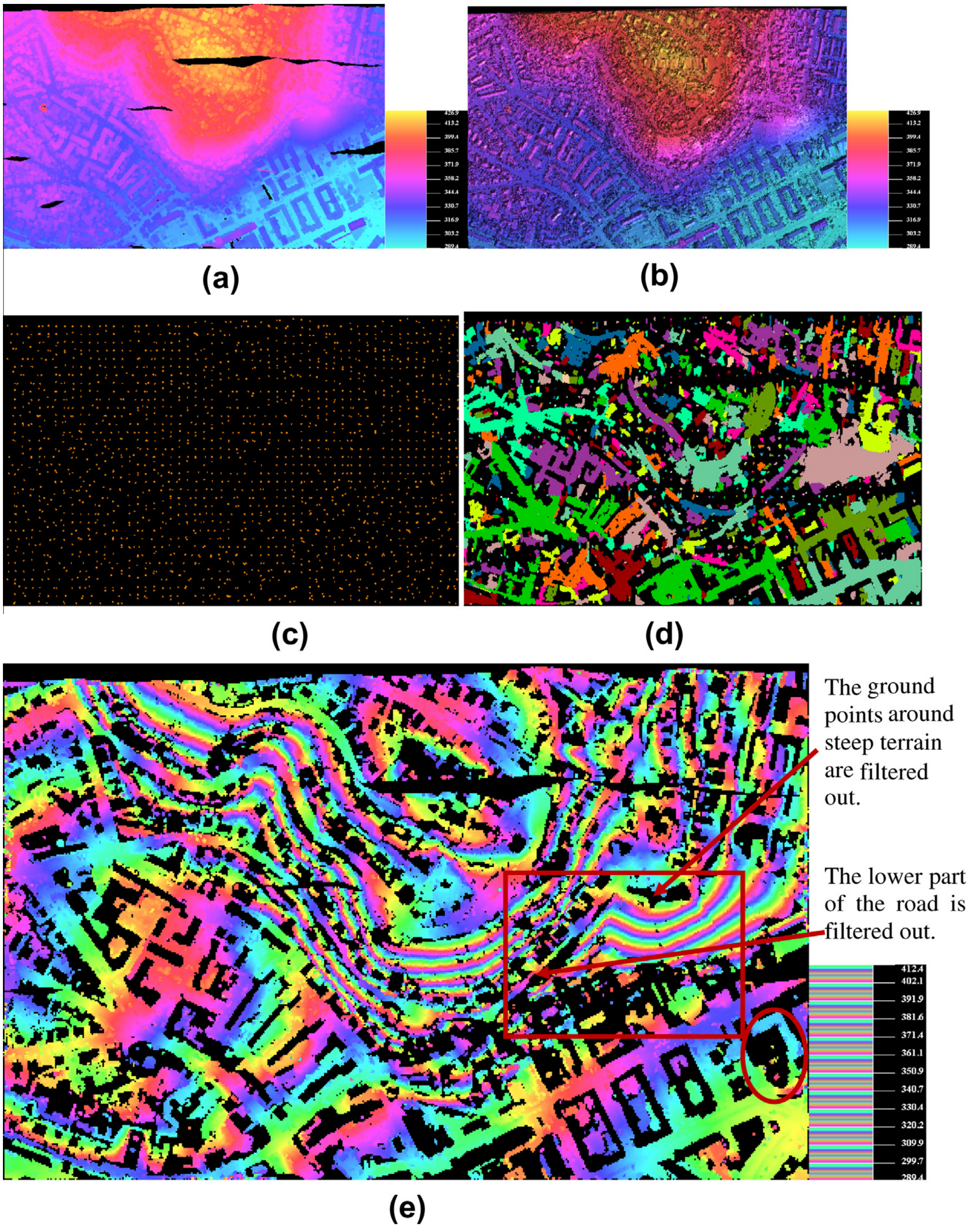
To compare the performance of the PTD method and our improved method, the ISPRS Commission III, Working Group III datasets are employed to test the filters. The ISPRS testing data were obtained by an Optech ALTM scanner over the Vaihingen/Enz test field and the Stuttgart city center. It includes eight sites consisting of different terrains: four urban sites and four rural/wooded sites, as well as 15 reference samples of sub-areas. The eight datasets are named as CSite1, CSite2, CSite3, CSite4, CSite5, CSite6, CSite7 and CSite8, respectively, as listed in Table 1. The overall characteristics of 8 data sets refer to Sithole and Vosselman (2004). As the CSite8 does not have a reference dataset, it was excluded for further experiment and analysis. The test data cover various land-use and land-cover types including buildings, vegetation, rivers, roads, railroads, bridges, etc. The laser data were collected with both first and last pulses recorded. The point spacing is 1–1.5 m for urban sites and 2–3.5 m for rural sites, respectively. Moreover, there are a total of 15 reference samples for testing the filtering accuracy. Particularly, the reference data were generated by manual filtering with knowledge of the seven landscapes and available aerial images (Sithole and Vosselman, 2004).

3.2. Specification of the parameters

As mentioned above, our proposed filtering algorithm needs two parameters more than the classic PTD method. In the following experiments, the shared five parameters are set to the same values for both of the filters, as shown in Table 1. All of the parameters are determined by the authors' experienced judgment on the conditions of the landscape, rather than by trial and error method, which is meaningful for the various types of applications. Table 1 summarizes the parameters used for sites 1–7. Note that all raw points are partitioned into filtering for the seven datasets, and the numbers of detected outliers refer to Table 1.

3.3. Results

With the specified parameters in Table 1, we perform the filtering on the seven datasets using the two methods. Among the filtering results, we select the ones of CSite1 and CSite2 as two



The ground points around steep terrain are filtered out.

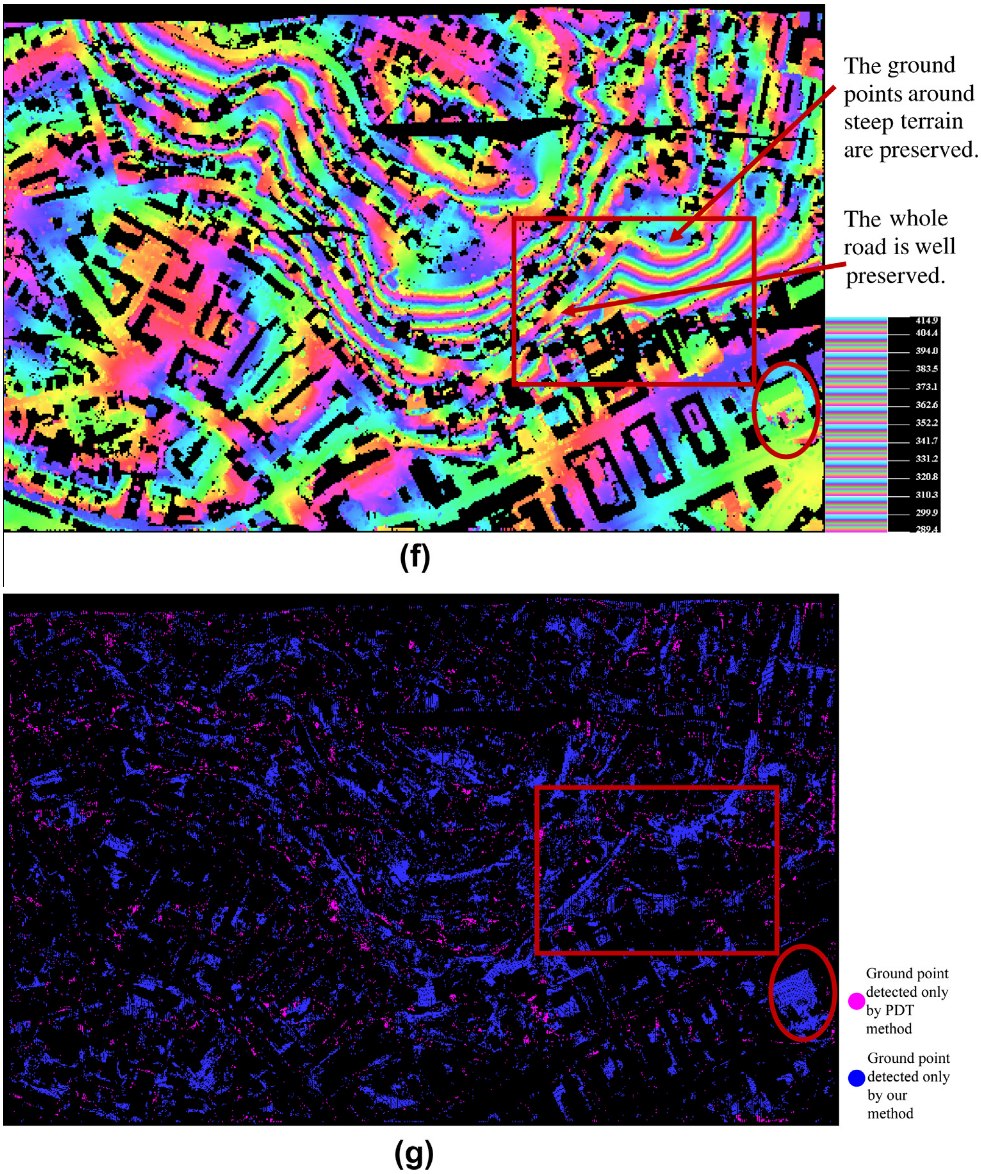
The lower part of the road is filtered out.

Fig. 5. Filtering and results of testing data about CSite1: (a) the remaining point cloud after outlier removal; (b) TIN of the data in (a); (c) ground seed points for the two filters; (d) detected ground measurements by the segmentation colored by the labeling number; (e) detected ground measurements by the classic PTD method; (f) detected ground measurements by our method; (g) differences between (e) and (f).

representatives to make a demonstration, as shown in Figs. 5 and 6, respectively. Furthermore, results of Sample11, Sample23, Sample42 and Sample53 are also displayed to reflect the details of filtering, as shown in Figs. 7–10, respectively.

CSite1 is located in an urban area, and its special features include steep slopes, mixture of vegetation and buildings on hillside, buildings on hillside, data gaps. There are 2,732,814 points in the raw data. 4970 points are identified as outliers and excluded from the remaining filtering process, and the remaining data and its TIN are displayed in Fig. 5a and b, respectively. During the third step, 1923 points are selected as ground seed points for the two filters, as shown in Fig. 5c. In the resulting point cloud, 996,071 points are detected as ground measurements for the PTD filter, as shown in Fig. 5e. In our method, 764,205 points are identified as ground points by the embedded SUSC algorithm, as shown in Fig. 5d; finally, there are 1,191,155 measurements being detected as ground, as shown in Fig. 5f. That means 64.16% of ground points have been detected by the SUSC before going into the fifth step. The differences of Fig. 5e and f are shown in Fig. 5g. Fig. 5g suggests that the main difference between the two results owns to the ability of our method to preserve the ground measurements in areas with steep terrain, as shown in the rectangular regions in Fig. 5e–g. Fig. 5e shows that most of the ground points around steep areas are omitted, and the lower part of a road across the steep terrain is missed for the classic PTD. In contrast, Fig. 5f shows that the ground measurements around the same steep areas and the whole road are well preserved by our method.

CSite2 is also located in an urban area, and its special features include large buildings, irregularly shaped buildings, a road with bridge and small tunnel, plus some data gaps. There are 973,598 points in the raw data. 124 points are identified as outliers. The remaining data without outliers and its TIN are displayed in


414.9
404.4
394.0
383.5
373.1
362.6
352.2
341.7
331.2
320.8
310.3
299.9
289.4
(f)
(g)
Ground point detected only by PDT method
Ground point detected only by our method

The ground points around steep terrain are preserved.

The whole road is well preserved.

Fig. 5. (*continued*)

Fig. 6a and b, respectively. During the third step, 70 points are selected as ground seed points for the two filters, as shown in Fig. 6c. In the resulting point cloud, 405,439 points are detected as ground measurements for the PTD method, as shown in Fig. 6e. In our method, 99,108 points are identified as ground points by the embedded SUSC algorithm, as shown in Fig. 5d. Finally, 483,491 points are detected as ground measurements, as shown in Fig. 6f, which means 20.50% ground points have been detected by the SUSC method before going into the fifth step. The differences of Fig. 6e and f are shown in Fig. 6g. The reason of the difference between the two results is our method's ability to preserve the ground measurement on some steep streets, as shown in the rectangular regions in Fig. 6e–g. Fig. 6e shows some road segments are omitted for the classic PTD. In contrast, Fig. 6f shows all of the roads are well preserved by our method.

Additionally, the detailed results from the references of Sample11, Sample23, Sample42 and Sample53, suggest that our proposed method can correctly detect more ground measurements than the PTD method in the case of:

- vegetation and buildings on a steep slope, as shown in Fig. 7;
- complex buildings, large buildings, break lines, as shown in Fig. 8;
- railway station with trains, as shown in Fig. 9;
- break lines, as shown in Fig. 10.
- However, our method fails to identify the object measurements under the following conditions:
- if the objects are closely attached to the underlying terrain surface, as the bridge shown in the ellipse region of Fig. 6f and g;
- if the objects are both small and close to the ground surfaces, as shown in Fig. 10f.

3.4. Performance evaluation

Both qualitative and quantitative assessments have been adopted to evaluate the performance of the two filters. Visual assessment of the filtered results of the seven datasets and 15 references shows that, both the classic PTD method and our method are robust to various types of complex landscapes such as data

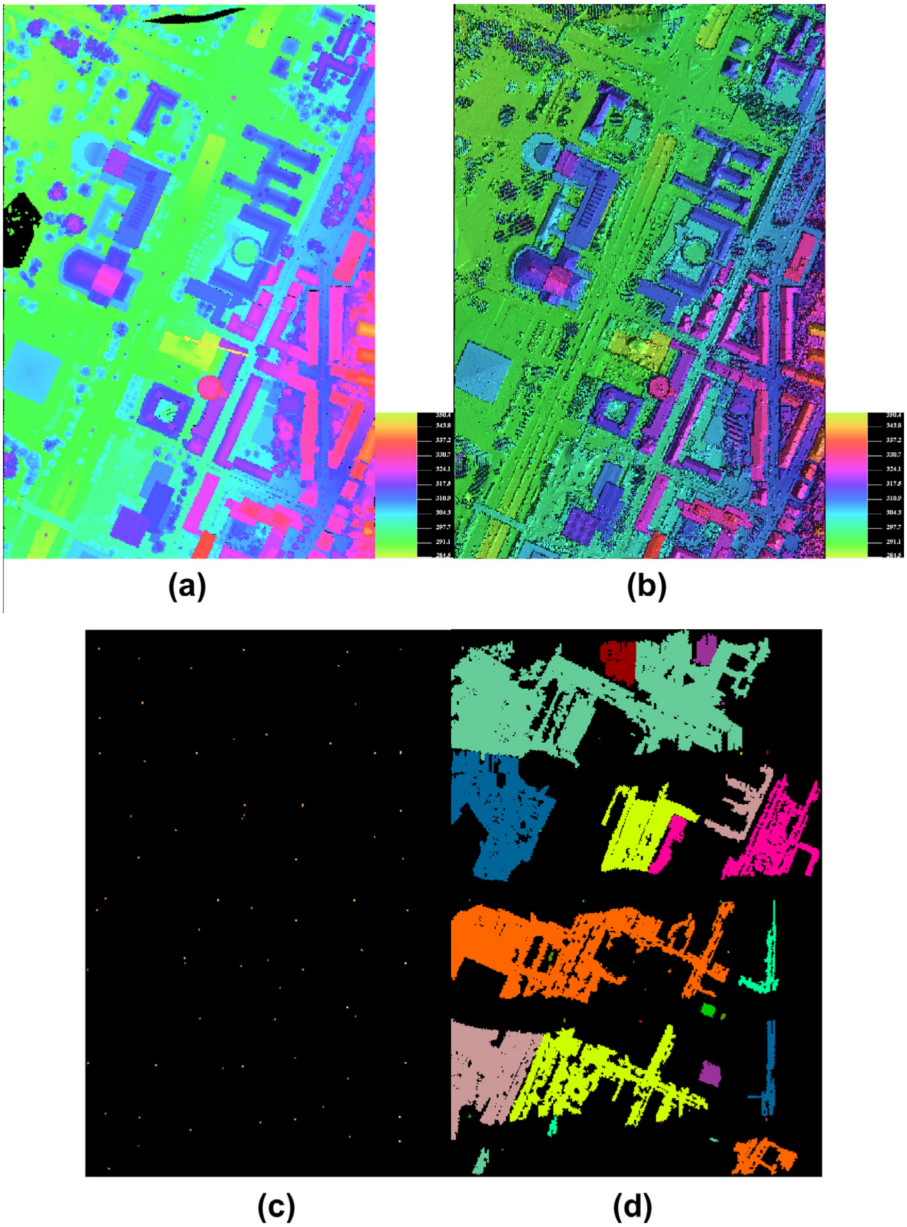


Fig. 6. Filtering and results of testing data about CSite2: (a) the remaining point cloud after outlier removal; (b) TIN of the data in (a); (c) ground seed points for the two filters; (d) detected ground measurements by the segmentation colored by the labeling number; (e) detected ground measurements by the classic PTD method; (f) detected ground measurements by our method; (g) differences between (e) and (f).

gaps, large buildings, irregularly shaped buildings, mixture of vegetation and buildings on flat terrains. However, the PTD algorithm fails to preserve the ground measurements in the cases of steep slopes, mixture of vegetation and buildings on a hillside, and just buildings on a hillside. Our proposed filter, however, is more likely to recognize the ground measurements when the PTD algorithm fails. On the other side, compared with the PTD algorithm, our method fails to identify the object measurements which are connected to the terrain surface through a smooth transition such as the very low buildings, as shown in the ellipse region in Fig. 5e–g, and the bridges, as shown in the ellipse region in Fig. 6e–g. The PTD algorithm would succeed when these conditions occur.

Additionally, we follow the quantitative assessment as proposed in ISPRS filter test (Sithole and Vosselman, 2004) to compare the two filters. Three kinds of errors are computed during the validation process, namely, type I errors (i.e., omission errors), type II errors (i.e., commission errors), and total errors. The former two types of errors are defined in Section 1, and the total error is the percentage of any misclassified points. The three types of errors of the two filters in the 15 references are listed in Table 2.

The statistics in Table 2 suggest that, both of the filters acquire high accuracies in the seven datasets, covering from urban areas to wooded areas, and the total error is less than 32.67% for all the filtered results, as shown in Fig. 11c. However, generally, our method has significantly lower type I error and total error than the PTD method. Specifically, among the 15 references, there are fourteen cases except the Sample21 where the type I errors of our method are lower than the PTD method, and there are twelve cases except the Sample21, Sample51, and Sample54 where the total error of our method are lower than the PTD approach, as shown in Fig. 11a and c. On average, compared to the PTD algorithm, the type I error and total error of our method are approximately reduced by 10.01% and 5.71%, respectively. However, the classic

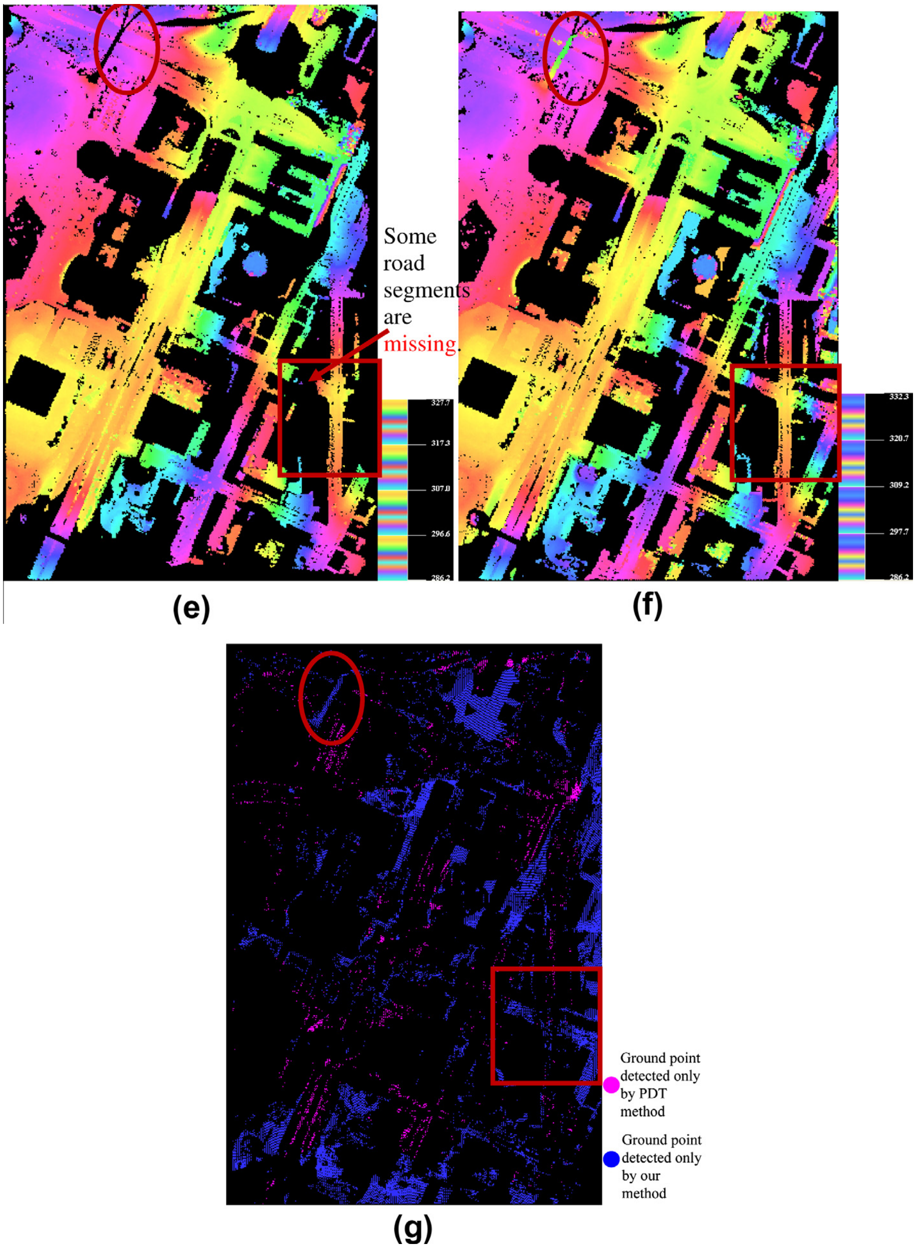


Fig. 6. (*continued*)

Table 1
Parameters of the two filters used for each site.

Scene	Parameters								
	Total number of points (points)	Number of outliers (points)	Classic PTD method					Our method	
			m (m)	t (°)	θ (°)	d (m)	l (m)	k (points)	α (°)
CSite1	2,732,814	4970	20	80.0	6.0	1.4	1.0	20	30.0
CSite2	973,598	124	60	88.0	6.0	1.4	1.0	20	10.0
CSite3	754,054	786	35	88.0	6.0	1.4	1.0	20	20.0
CSite4	1,036,114	5128	60	88.0	6.0	1.4	1.0	25	30.0
CSite5	1,256,894	398	10	70.0	6.0	1.0	2.0	20	30.0
CSite6	1,101,952	4444	40	70.0	6.0	1.4	2.0	20	30.0
CSite7	786,134	2172	20	70.0	6.0	1.4	2.0	30	30.0

PTD approach also has its advantage in avoiding type II errors. The statistics in Table 2 tell us that, there are ten cases where our proposed method has higher type II errors than the PTD algorithm, as shown in Fig. 11b. However, the above disadvantage of our method is not fatal. Considering that our method is likely to have lower type I errors and total errors, our method will need less human involvements compared to the PTD method, because the cost of repairing the type II errors is far lower than the ones of repairing

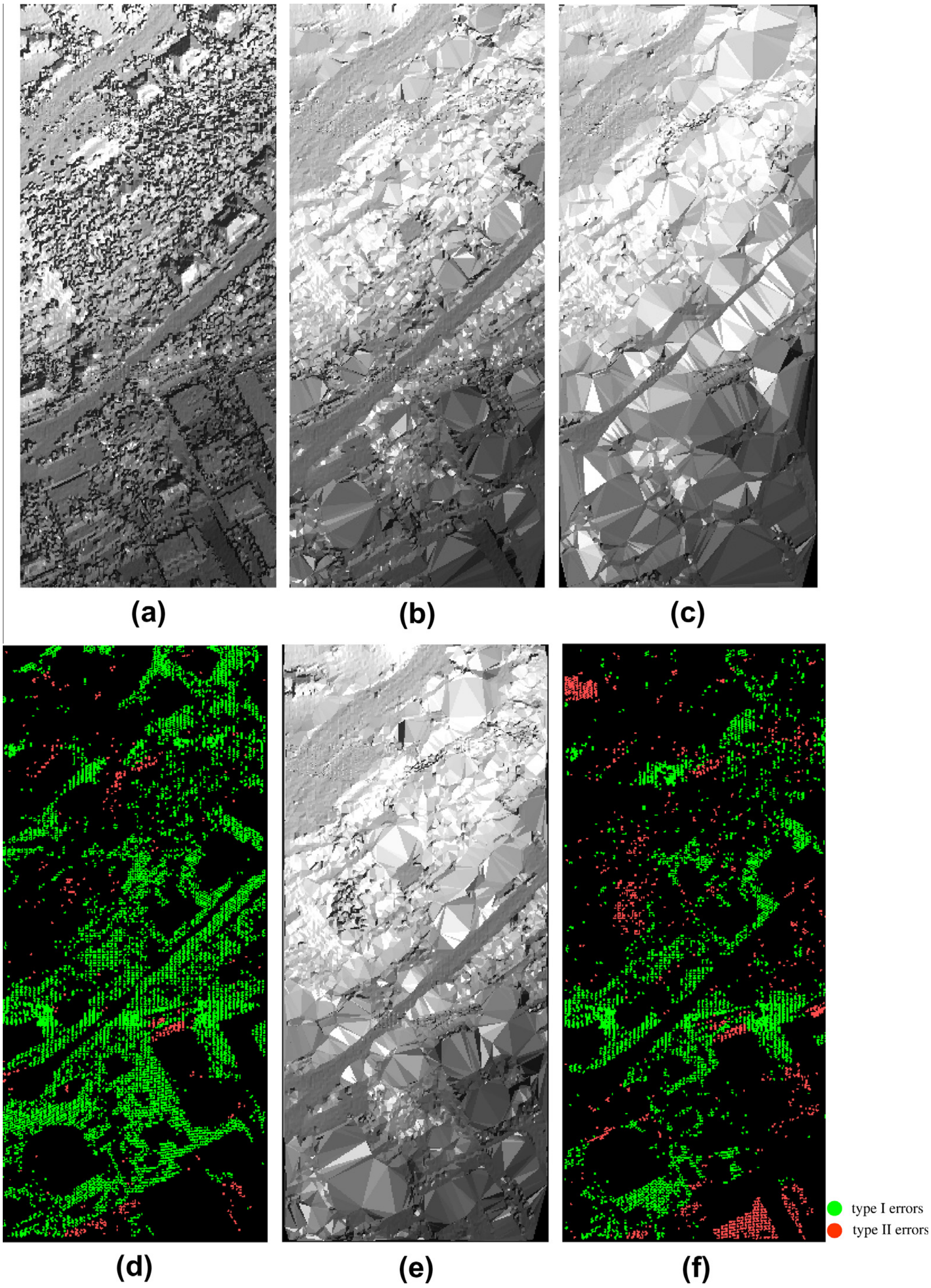


Fig. 7. Filtering and results of reference data of Sample11: (a) the digital surface model (DSM); (b) the reference DEM; (c) the DEM generated from the result of the PTD method; (d) the type I errors, type II errors of the PTD method; (e) the DEM generated from the result of our method; (f) the type I errors, type II errors of our method.

the type I error in the stage of manual operation after automatic filtering (Sithole and Vosselman, 2004).

Another disadvantage of our proposed approach is that it needs two more specified parameters, namely k and α, needed for the segmentation. Based on the scene complexity and statistics in Table 1, CSite1 and Sample11 are selected to analyze the sensitivity of the two parameters and the effect on three types of errors, and the results are shown in Fig. 12. When $k \in [10, 30]$ and the other parameters are constant, the type I errors decrease from 48% into 18%, the type II errors increase from 3% into 14%, and the total errors decrease from 29% into 17%. The above obvious difference owns to the sensitivity of k to the plane fitting in SUSC if k is not large enough. However, when k is larger than 18, the three types will not change significantly. Actually, when $k \in [18, 30]$, the type I errors decrease from 27% into 18%, the type II errors increase from 9% into 14%, the total errors decrease from 19% into 17%, despite there is also a slight fluctuation when $k = 24$ for the type I error. Similarly, when $\alpha \in [16°, 36°]$ and the other parameters are constant, the type I errors decrease from 34% into 27%, the type II errors increase from 6% into 8%, and the total errors decrease from 22% into 19%. In a word, the three types are not sensitive to the change of values of α. Thus, we can get similar results if we chose

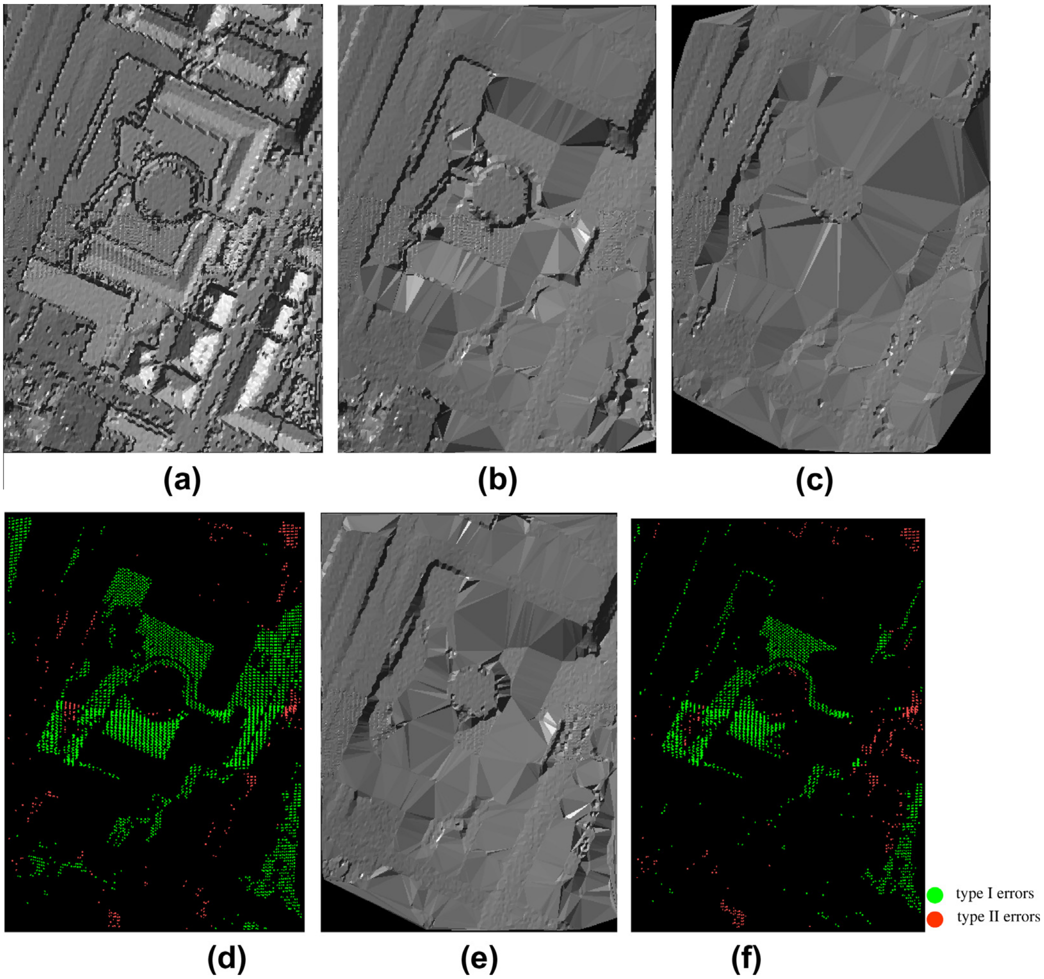


Fig. 8. Filtering and results of reference data of Sample23: (a) The DSM; (b) The reference DEM; (c) The DEM generated from the result of the PTD method; (d) The type I errors, type II errors of the PTD method; (e) The DEM generated from the result of our method; (f) The type I errors, type II errors of our method.

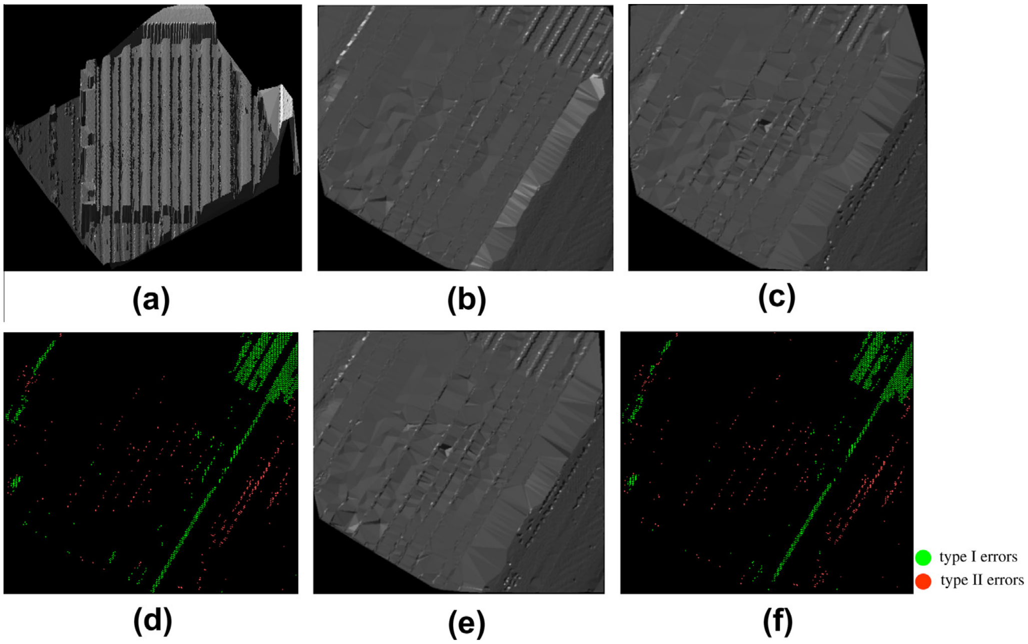


Fig. 9. Filtering and results of reference data of Sample42: (a) the DSM; (b) the reference DEM; (c) the DEM generated from the result of the PTD method; (d) the type I errors, type II errors of the PTD method; (e) the DEM generated from the result of our method; (f) The type I errors, type II errors of our method.

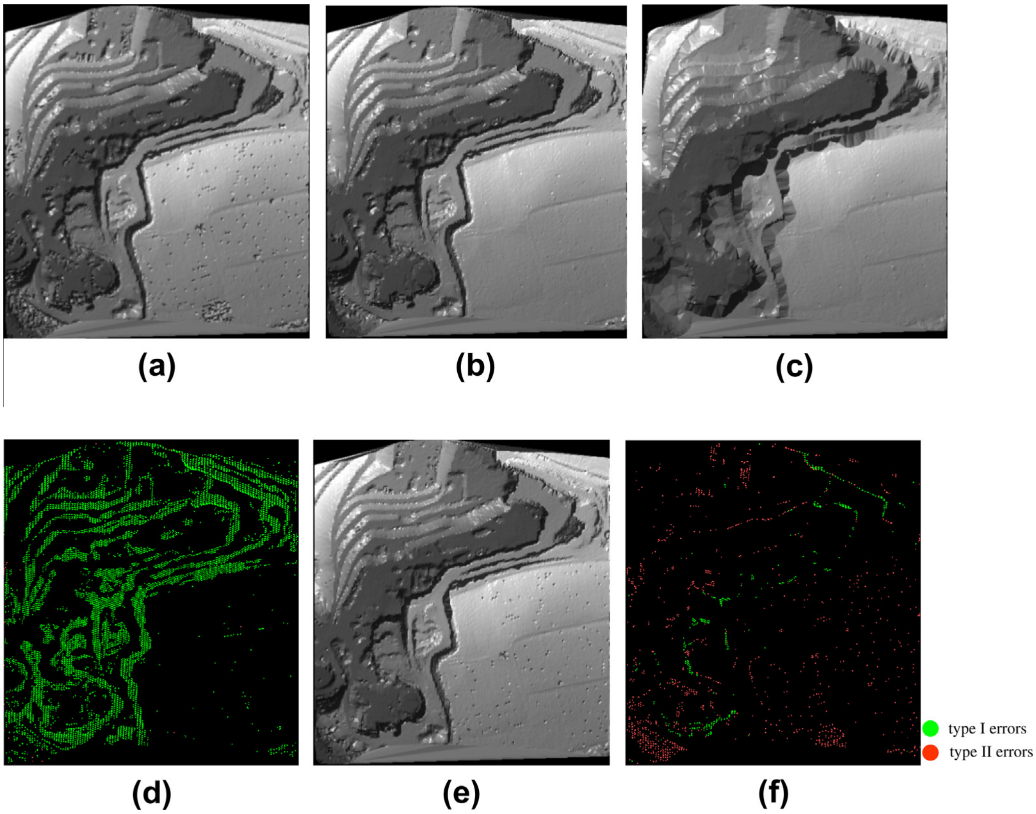
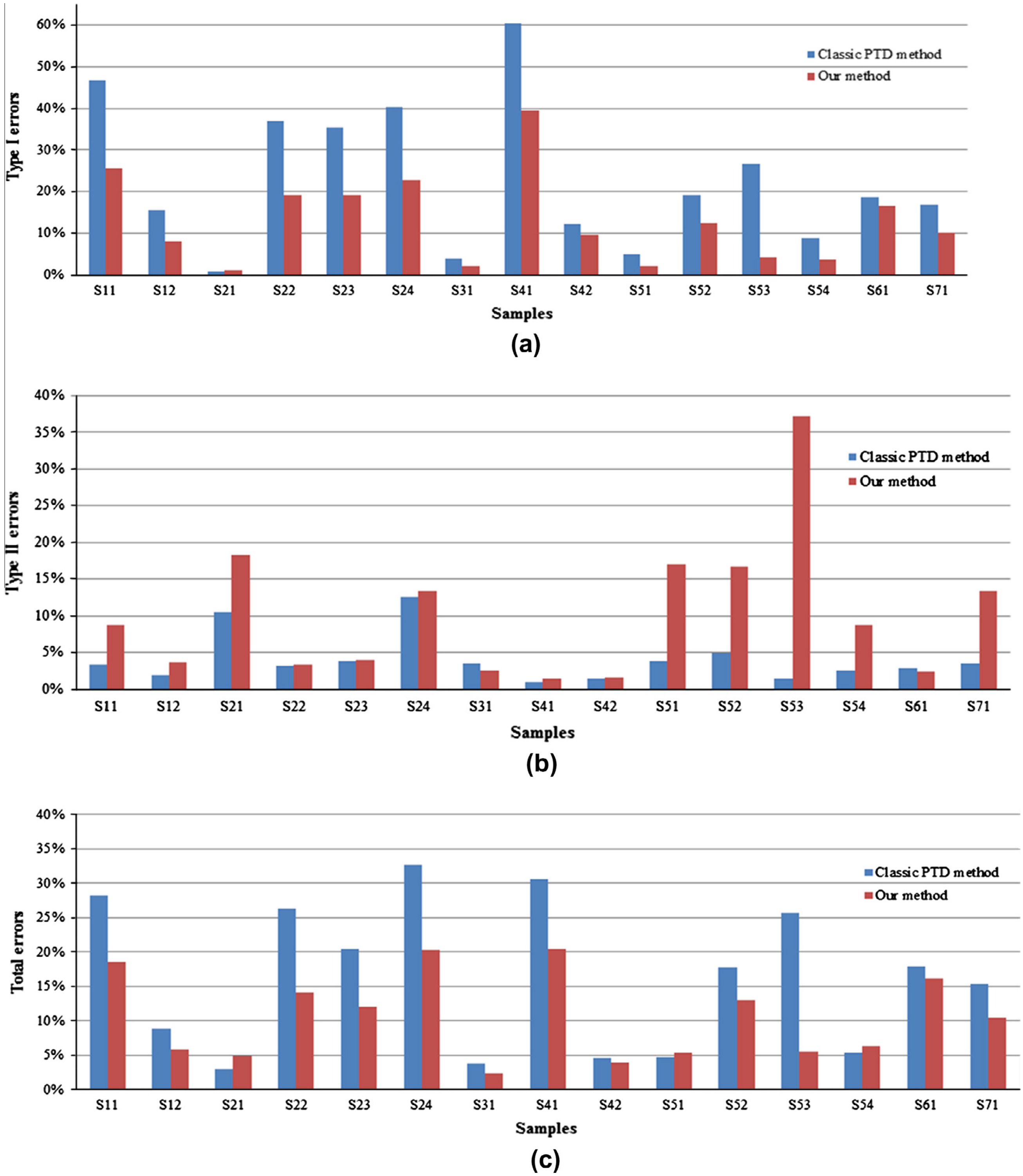


Fig. 10. Filtering and results of reference data of Sample53: (a) the DSM; (b) the reference DEM; (c) the DEM generated from the result of the PTD method; (d) The type I errors, type II errors of the PTD method; (e) the DEM generated from the result of our method; (f) the type I errors, type II errors of our method.

Table 2
Three types of errors of the two filters, i.e., PTD method and our method.

Dataset no.	Type of error	PTD (%)	Ours (%)	Dataset NO. /Types	Type of error	PTD (%)	Ours (%)
Sample11	I	46.68	25.67	Sample51	I	4.91	2.05
	II	3.4	8.84		II	3.8	16.97
	T	28.21	18.49		T	4.67	5.31
Sample12	I	15.6	8.13	Sample52	I	19.2	12.53
	II	1.92	3.61		II	4.95	16.77
	T	8.93	5.92		T	17.7	12.98
Sample21	I	0.78	1.17	Sample53	I	26.66	4.25
	II	10.47	18.23		II	1.44	37.22
	T	2.93	4.95		T	25.64	5.58
Samp22	I	36.84	19.05	Sample54	I	8.76	3.59
	II	3.23	3.44		II	2.53	8.82
	T	26.36	14.18		T	5.41	6.4
Sample23	I	35.33	19.25	Sample61	I	18.52	16.62
	II	3.82	4.05		II	2.82	2.49
	T	20.42	12.06		T	17.98	16.13
Sample24	I	40.3	22.86	Sample71	I	16.81	10.07
	II	12.54	13.41		II	3.5	13.39
	T	32.67	20.26		T	15.3	10.44
Sample31	I	3.93	2.1	Minimum	I	0.78	1.17
	II	3.55	2.59		II	0.91	1.44
	T	3.76	2.32		T	2.93	2.32
Sample41	I	60.34	39.54	Maximum	I	60.34	39.54
	II	0.91	1.44		II	12.54	37.22
	T	30.55	20.44		T	32.67	18.49
Sample42	I	12.13	9.72	Average	I	23.12	13.11
	II	1.45	1.55		II	4.02	10.19
	T	4.58	3.94		T	16.34	10.63

slightly different values. The Fig. 12 also suggests that we have not fine-tuned these particular examples so the results come out favorably, but just made decisions on experiences, because $k = 20$ is not the optimal result in view of three types of errors. From the parameters in Table 1, we conclude that k is greatly dependent on the point density of the point clouds and the landscape complexity,



(a)

(b)

(c)

Fig. 11. Comparison of the three types of errors for the two methods: (a) type I errors of different algorithms for the samples; (b) type II errors of different algorithms for the samples; (c) total errors of the two algorithms for the samples. Note “S” is abbreviation of “Sample” for the horizontal captions.

and α is greatly dependent on the maximum slope of the terrain and acquisition errors in the original point cloud data. Particularly, a value of 20 for k and values from 10° to 30° for α turned out to be feasible for most of the test datasets. In other words, the additional two parameters do not significantly add complexity to our proposed method. From the above statistics and analysis, we conclude that our method has a significant reduction of the type I errors and total errors compared to the PTD method without significantly adding to the complexity of the algorithm, thus it is more practical than the PTD method.

A further disadvantage of our method is that it needs more computation time compared to the PTD method, because of the segmentation and the insertion of its results into the TIN, as illustrated in Section 2.3. As far as the time cost is concerned, it takes 48.906, 18.093, 15.141, 20.25, 36.188, 25.578, 18.891 s to finish the filtering tasks of CSite1-CSite7 respectively for the PDT method.

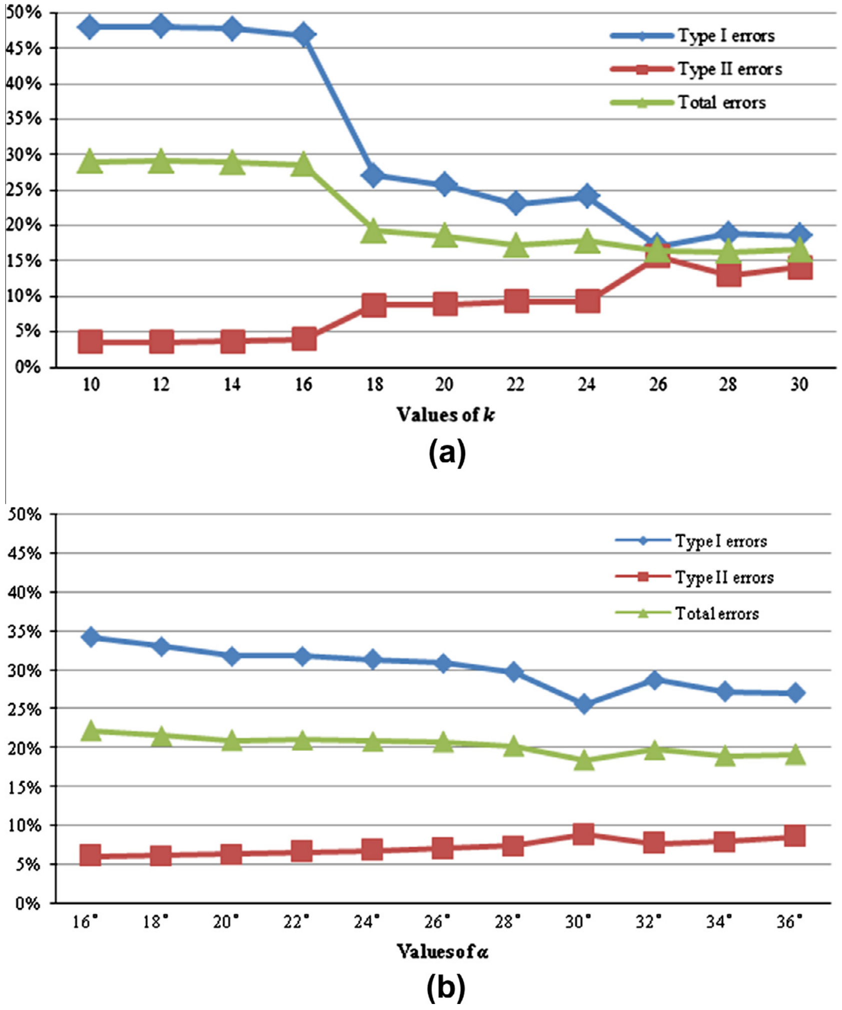


Fig. 12. Analysis on sensitivity of the input parameters k and α respectively, and its effect on the three types of errors for the Sample11: (a) values of k and the corresponding three types of errors when $\alpha = 30°$; (b) values of α and the corresponding three types of errors when $k = 20$.

Correspondingly, the time cost is 594.828, 88.406, 72.109, 103.281, 378.089, 174.688, 118.328 s respectively for our proposed method. On average, time cost of our method is 8.36 times of the one of the PDT method on our computer. However, this paper focuses on a filter's accuracy rather than efficiency, and the efficiency of a filter method is an easier problem to solve if parallel computing is considered.

The advantages of our method result from the following two factors. The first one is the adopting of the point cloud segmentation in the process of filtering. Particularly, the SUSC method will expand the set of initial ground points to a large extent if the natural terrain is smooth enough, as shown in both of Figs. 5d and 6d. As a result, the increased number of ground seed points reduces the possibility of omitting the remaining ground measurements for our filtering algorithm. The last but not least factor is the inheritance of the flow chart of classic PTD method. As mentioned above, the PTD method is an excellent filter, and it has been widely applied due to the popularization of the TerraSolid commercial software. Our approach makes best use of the flow chart of the classic PTD algorithm, and only makes some improvement at some steps, which yields the better performance of our method. However, embedding the SUSC into the PTD is also a double-edged sword. The point cloud segmentation also makes our method more likely regard some object points attached to the terrain as ground points, which probably increases the type II errors for our method, as shown in Table 2 and Fig. 10f.

4. Conclusions and discussion

Filtering is one of the core post-processing steps for ALS point clouds, and many filters have been proposed to solve this problem. Among them, PTD is widely applied as one of the surface-based ones. However, there are no details reported about this filter due to the protection of proprietary work, therefore we implemented the PTD from scratch. However, the PTD fails to preserve the ground measurements in steep terrain areas, even if the mirroring technique is adopted. Thus, the classic filtering method is improved by the SUSC algorithm. The SUSC algorithm has the capability of expanding the ground seed surfaces into surrounding smooth terrains as much as possible. As a result, the SUSC is helpful to derive more new ground seed points, which is useful to the subsequent densification of the TIN surfaces. The experiments make use of the 7 datasets of ISPRS Commission III/Working Group III to verify our proposed method; moreover, the 15 reference samples from the sub-areas are used to calculate the accuracies of the proposed approach. The results suggest that both our method and the classic PTD are robust to various types of landscapes. However, our proposed approach is better than the classic PTD method in preserving the ground measurements. Particularly, it may have significant lower type I errors and total errors than the PTD algorithm despite that it may have higher type II errors, which will reduce the costs of the following manual operations. However, our proposed method may fail when it is faced with objects which are attached to the ground, such as bridges, ramps, etc. The future work will focus on the improvement of the proposed filter to reduce the type II errors.

Acknowledgments

The authors thank the anonymous reviewers for their insightful comments and suggestions. This research was funded by the Project for Young Scientist Fund sponsored by the Natural Science Foundations of China (NSFC) under Grant 41001280, the China Postdoctoral Science Foundation under Grant 2010047038, and Open Foundations of Key Laboratory of Mapping from Space of National Administration of Surveying, Mapping and Geoinformation under Grant K201003, respectively.

References

Arefi, H., Hahn, M., 2005. A morphological reconstruction algorithm for separating off-terrain points from terrain points in laser scanning data. International Archives of Photogrammetry, Remote Sensing and Spatial, Information Sciences 36 (Part3/W19), 120–125.

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y., 1998. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. Journal of the ACM 45, 891–923.

Axelsson, P.E., 2000. DEM generation from laser scanner data using adaptive TIN models. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (Part B4/1), 110–117.

Chen, H., Cheng, M., Li, J., 2012. An iterative terrain recovery approach to automated DTM generation from airborne LiDAR point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial, Information Sciences XXXIX-B4, 364–368.

Chen, Q., Gong, P., Baldocchi, D., Xie, G., 2007. Filtering airborne laser scanning data with morphological methods. Photogrammetric Engineering and Remote Sensing 73 (2), 175–185.

Filin, S., Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS Journal of Photogrammetry and Remote Sensing 60 (2), 71–80.

Hyyppä, J., Schardt, M., Haggrén, H., Koch, B., Lohr, U., Scherrer, H.U., Paananen, R., Luukkonen, H., Ziegler, M., Hyyppä, H., Pyysalo, U., Friedländer, H., Uuttera, J., Wagner, S., Inkinen, M., Wimmer, A., Kukko, A., Ahokas, E., Karjalainen, M., 2001. HIGH-SCAN: the first European-wide attempt to derive single tree information from laser scanner data. The Photogrammetric Journal of Finland 17, 58–68.

Kilian, J., Haala, N., Englich, M., 1996. Capture and evaluation of airborne laser scanner data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 31 (Part B3/W1), 385–388.

Koch, B., Heyder, U., Weinacker, H., 2006. Detection of individual tree crowns in airborne LiDAR data. Photogrammetric Engineering and Remote Sensing 72 (4), 357–363.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing 53 (4), 193–203.

Maas, H.G., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing 54 (2–3), 153–163.

Melzer, T., 2007. Non-parametric segmentation of ALS point clouds using mean shift. Journal of Applied Geodesy 1 (3), 159–170.

Meng, X., Currit, N., Zhao, K., 2010. Ground filtering algorithms for airborne LiDAR data: a review of critical issues. Remote Sensing 2 (3), 833–860.

Meng, X., Wang, L., Silvan, J., Currit, N., 2009. A multi-directional ground filtering algorithm for airborne LIDAR. ISPRS Journal of Photogrammetry and Remote Sensing 64 (1), 117–124.

Mount, D.M., Arya, S., 2010. ANN: A Library for Approximate Nearest Neighbor Searching, <http://www.cs.umd.edu/~mount/ANN> (accessed 10.09.10).

Oude Elberink, S., Vosselman, G., 2009. 3D information extraction from laser point clouds covering complex road junctions. Photogrammetric Record 24 (125), 23–36.

Pfeifer, N., Stadler, P., Briese, C., 2001. Derivation of digital terrain models in the SCOP++ environment. In: Proceedings of OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Digital Elevation Models, Stockholm, Sweden (on CD-ROM).

Rabbani, T., van den Heuvel, F.A., Vosselman, G., 2006. Segmentation of point clouds using smoothness constraints. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 5), 248–253.

Shan, J., Sampath, A., 2005. Urban DEM generation from raw Lidar data: a labeling algorithm and its performance. Photogrammetric Engineering and Remote Sensing 71 (2), 217–226.

Shao, Y., Chen, L., 2008. Automated searching of ground points from airborne LiDAR data using a climbing and sliding method. Photogrammetric Engineering and Remote Sensing 74 (5), 625–635.

Shen, J., Liu, J.P., Lin, X.G., Zhao, R., 2012. Object-based classification of airborne light detection and ranging point clouds in human settlements. Sensor Letters 10 (1–2), 221–229.

Shewchuk, J.R., 2005. Triangle: A Two-dimensional Quality Mesh Generator and Delaunay Triangulator. <http://www.cs.cmu.edu/~quake/triangle.html> (accessed 10.10.11).

Sithole, G., 2005. Segmentation and Classification of Airborne Laser Scanner Data, vol. 59. Publications on Geodesy of the Netherlands Commission of Geodesy, Delft.

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59 (1–2), 85–101.

Soille, P., 2003. Morphological Image Analysis: Principles and Applications, second ed. Springer-Verlag, New York, USA.

Terrasolid Ltd., 2010. TerraScan User's Guide. <http://www.terrasolid.fi/en/users_guide/terrascan_users_guide> (accessed 08.10.11).

Vosselman, G., 2009. Advanced point cloud processing. In: Fritsch, D. (Ed.), Photogrammetric Week. Wichmann Verlag, Heidelberg, Germany, pp. 137–146.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognizing structure in laser scanner point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial, Information Sciences 36 (Part 8/W2), 33–38.

Vosselman, G., Klein, R., 2010. Visualization and structuring of point clouds. In: Vosselman, G., Maas, H.G. (Eds.), Airborne and Terrestrial Laser Scanning. Whittles Publishing, USA, pp. 43–79.

Wang, M., Tseng, Y.H., 2010. Automatic segmentation of LiDAR data into coplanar point clusters using an octree-based split-and-merge algorithm. Photogrammetric Engineering and Remote Sensing 76 (4), 407–420.

Zhang, J.X., 2010. Multi-source remote sensing data fusion: status and trends. International Journal of Image and Data Fusion 1 (1), 5–24.

Zhang, K.Q., Chen, S.C., Whitman, D., Shyu, M.L., Yan, J.H., Zhang, C.C., 2003. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. IEEE Transactions on Geoscience and Remote Sensing 41 (4), 872–882.