Optics and Laser Technology 112 (2019) 43–55

Contents lists available at ScienceDirect

Optics and Laser Technology

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

Full length article

Automatic DTM extraction from airborne LiDAR based on expectation-maximization

Zhenyang Hui[a,e,*], Dajun Li[a,*], Shuanggen Jin[b], Yao Yevenyo Ziggah[c], Leyang Wang[a], Youjian Hu[d]

[a] Faculty of Geomatics, East China University of Technology, Nanchang 330013, China
[b] Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China
[c] Faculty of Mineral Resources Technology, University of Mines and Technology, Tarkwa, Ghana
[d] Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China
[e] Key Laboratory for Digital Land and Resources of Jiangxi Province, East China University of Technology, Nanchang 330013, China

HIGHLIGHTS:

- Filtering is seen as a separation of mixed Gaussian models.
- Ground points can be extracted automatically by applying the EM algorithm.
- The fitted surface can be generated using ground seeds.
- Revised elevations can enhance the robustness of the filtering algorithm.

ARTICLE INFO

Keywords:
Airborne LiDAR
Point clouds
Filtering
Expectation-maximization
Least-squares fitting

ABSTRACT

Filtering of ground points is a key step for most applications of airborne LiDAR point clouds. Although many filtering algorithms have been proposed in recent years, most of them suffer from parameter setting or thresholds fine-tuning. This is most often time-consuming and reduces the degree of automation of the applied algorithm. To overcome such problems, this paper proposes a threshold-free filtering algorithm based on expectation–maximization (EM). The filter is developed based on the assumption that point clouds are seen as a mixture of Gaussian models. Thus, the separation of ground points and non-ground points from point clouds is partitioning of the point clouds by a mixed Gaussian model that is used for screening ground points. EM is applied to realize the separation, which calculates the maximum likelihood estimates of the mixture parameters. Using the estimated parameters, the likelihoods of each point belonging to ground or non-ground are computed. Noticeably, point clouds are labeled as the component with a larger likelihood. The proposed method has been tested using the standard filtering datasets provided by the ISPRS. Experimental results showed that the proposed method performed the best in comparison with the classic progressive triangulated irregular network densification (PTD) and segment-based PTD methods in terms of omission error. The average omission error of the proposed method was 52.81% and 16.78% lower than the classic PTD method and the segment-based PTD method, respectively. Moreover, the proposed method was able to reduce its average total error by 31.95% compared to the classic PTD method.

1. Introduction

Airborne light detection and ranging (LiDAR) applications have been developing very quickly in recent decades. Compared with traditional remote sensing methods, airborne LiDAR can collect geographic information efficiently [1,2]. Moreover, weather conditions have no influence on this technique when gathering point clouds [3]. Thus, airborne LiDAR has been widely used in many areas, such as digital terrain model (DTM) extraction [4,5], three-dimensional building model generation [6,7], road extraction [8,9], forest parameter estimation [10,11], etc.

Although airborne LiDAR is cost-effective when acquiring three-

* Corresponding authors at: Faculty of Geomatics, East China University of Technology, Nanchang 330013, China (Z. Hui).
E-mail addresses: huizhenyang2008@ecit.cn (Z. Hui), djli@ecit.cn (D. Li), sgjin@shao.ac.cn (S. Jin), yyziggah@umat.edu.gh (Y.Y. Ziggah).

https://doi.org/10.1016/j.optlastec.2018.10.051
Received 1 June 2018; Received in revised form 15 October 2018; Accepted 22 October 2018
0030-3992/

dimensional information, processing point cloud data requires considerable time, especially in manual classification. It has been established that the manual classification and quality control consume an estimated 60–80% of processing time [12]. Thus, it is urgent to develop algorithms to speed up point clouds post-processing efficiency.

To develop algorithms for many other applications, one fundamental step is extracting DTM information from point clouds that contain both ground points and non-ground points [13]. This process is generally known as filtering. Aiming at realizing filtering effectively, lots of algorithms have been put forward in the past twenty years. These filtering algorithms can be categorized into four classes: slope-based, morphology-based, surface-based, and segmentation-based [1,2,14].

The slope-based approaches always assume that the gradients between non-ground points and ground points are larger than those between ground points. Thus, if the slope or height difference between two points is larger than a pre-defined threshold, the point with higher elevation will be considered as a non-ground point. Vosselman first proposed the slope-based algorithm [15]. Many researchers have proposed some modified algorithms based on this pioneering work to make the traditional method adaptable to complex terrain environments [16–18]. However, the experimental results show that the performance of this kind of approach is sensitive to the threshold setting, and when encountering complicated terrains, the slope-based approaches cannot achieve satisfactory accuracy [19,20].

In morphology-based approaches, some morphology operations, namely, dilation, erosion, opening and closing are involved. To realize this kind of method, the key issue is to choose an appropriate window size. A large window size will flatten the terrain details, while a small window size has no effects on filtering large building roofs. To overcome this problem, Zhang et al. [21] proposed a progressive morphological filtering method by gradually changing the window size and threshold. Nonetheless, there are two main problems with this algorithm. First, this method assumes the slope of the entire terrain as a constant, which is obviously unreasonable in undulation environments. Second, this method cannot effectively protect the terrain details. To solve these problems, many modified variants of this traditional work have been proposed [22–26]. Recently, Hui et al. [27] improved the progressive morphological filter by combining it with a multi-level interpolation filtering method. Promising results were achieved in complicated terrain environments.

The basic idea of surface-based approaches is to build approximate terrain surfaces progressively using certain interpolation methods. By setting some filtering rules, such as residuals to the interpolated surface, the non-ground points are eliminated iteratively. Kraus and Pfeifer [28] adopted a linear prediction method and a weight function to filter the non-ground points step-by-step. A promising result was obtained in wooded areas. Axelsson [29] first proposed a progressive triangular irregular network (TIN) densification (PTD) approach. In this method, a coarse TIN is built initially using some ground seeds with the lowest elevation in local areas. Then ground points are added to the TIN iteratively if they satisfy certain criteria, including iteration distance and angle. The iteration ends when no more points can be added. Hu et al. [30] adopted thin plate spline (TPS) interpolator to construct a raster surface. In their method, the bending energy is calculated as a byproduct of TPS interpolation, which is used to calculate the adaptive threshold automatically. In doing so, the method does not need cautious parameter tuning and can yield a better filtering result when optimized.

The segmentation-based approaches always involve two steps: first, point clouds are segmented based on some methods, namely, 3D Hough transform [31,32], random sample consensus (RANSAC) [33], region growing [20,34], scan line segmentation [35,36], etc. Second, certain rules, such as smoothness or height difference, are applied to discriminate terrain segments from clusters generated in the last step. Because the segmentation results often contain more semantic information, many researchers have developed hybrid models that integrate the segmentation methods with some traditional filtering methods to improve the filtering accuracy. For example, Tovari and Pfeifer [37] adopted a region growing algorithm to modify the linear-prediction method proposed by Kraus and Pfeifer. The experiments show that the improved method decreases the filtering errors at break lines. Lin and Zhang [20] further developed the PTD algorithm by first applying the point cloud smooth segmentation method. In this approach, a segment, rather than a single point, is processed as a unit. Thus, the proposed technique preserves discontinuities of landscapes. Moreover, some lower objects attached to the ground can also be removed successfully. Recently, Chen et al. [38] improved their original multi-resolution hierarchical classification (MHC) algorithm by first segmenting the LiDAR points using the region growing method. The experimental results show that the improved MHC is more accurate than the original MHC, irrespective of the accuracy measures.

Although most of the proposed methods in the literature yield good filtering performance, they still require complicated parameter tuning when encountering various types of terrain, such as urban areas, mountainous areas, forested areas, etc. Parameter-tuning is generally time consuming and always incurs heavy manual editing costs. Thus, these algorithms are not easy for inexperienced users to realize filtering. To overcome this problem, this paper proposes a parameter-free filtering algorithm based on expectation–maximization (EM). The proposed algorithm is developed based on the assumption that point clouds are seen as a mixture of Gaussian models. The separation of ground points and non-ground points from point clouds partitions the point clouds by a mixed Gaussian model that is used for screening ground points. EM is applied for realizing the separation. The proposed method is tested using the datasets provided by the International Society for Photogrammetry and Remote Sensing (ISPRS). Experimental results show that the proposed EM method achieves good performance under variant terrain features without any human manipulation. Thus, the proposed method will be user-friendly, thereby providing a good foundation for the automatic algorithm development of post-processing applications for airborne LiDAR.

2. Methodology

According to the central limit theorem, naturally measured LiDAR data will lead to a normal distribution [39,40]. Conversely, due to the complex terrain environments, point clouds can be assumed as a mixture of Gaussian models. Therefore, the separation of ground points and non-ground points from point clouds can be seen as a separation of a mixed Gaussian model. EM is an approach for fitting probability distributions and calculates the maximum likelihood estimates of parameters to probabilistic models being fit to the data. When we do not know which component (ground or non-ground) the point belongs to, EM can be used to calculate maximum likelihood estimates of the mixture parameters. Using the estimated parameters, the likelihood of each point belonging to ground or non-ground can be computed. It is obvious that the point is labeled as the class corresponding to the maximum likelihood. Fig. 1 describes the flow chart of the proposed method. The detailed procedures are presented in Table 1, which comprise the following three steps: i. outlier removal, ii. revised elevation calculation, and iii. ground point extraction using EM.

2.1. Outlier removal

Due to the influence of external environments or the laser rangefinder malfunction, the acquired point clouds always contain noisy points, including high and low outliers. Both of these outliers may disturb the assumed normal distribution; in particular, the low outliers may have a large influence on the final filtering results. This is because many filtering algorithms assume that the lowest points belong to ground points. Both high and low outliers can be removed based on the fact that their elevations are commonly abrupt compared to their neighbors.

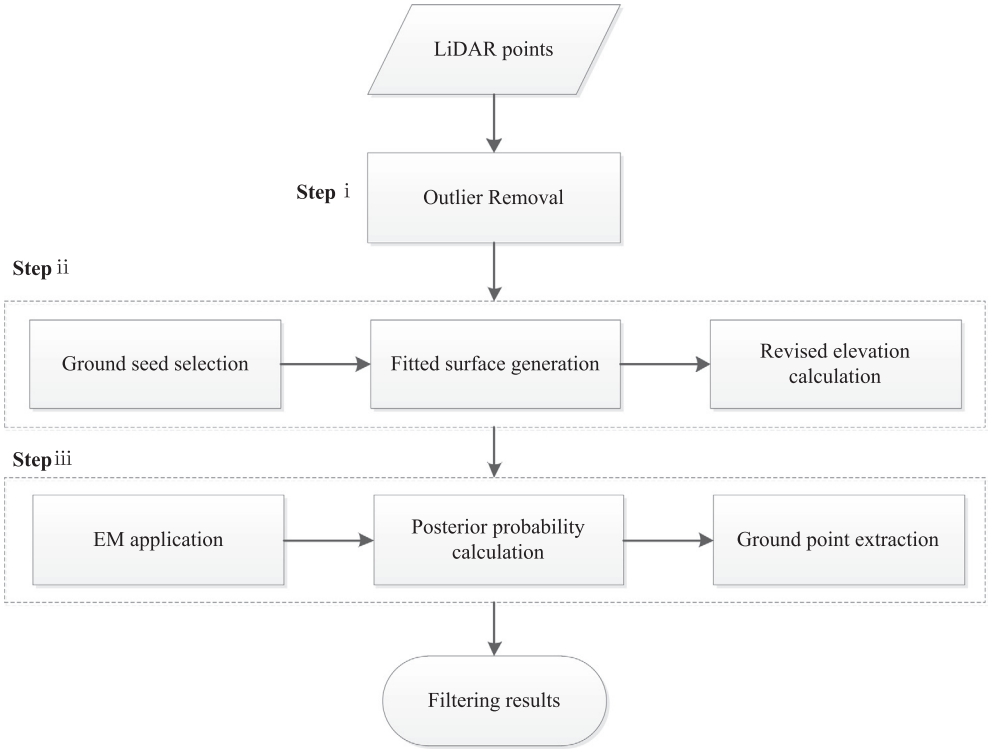

Fig. 1. Flow chart of the proposed method.

Table 1
Algorithmic steps of the proposed method.

Input: point cloud data including X, Y, Z coordinates
Output: ground points after filtering

Step i: Outlier removal
1. Point clouds organized by a *k*-dimensional tree.
2. K nearest neighbors found.
3. Outlier removal according to Equations (1)–(4).

Step ii: Revised elevation calculation
1. Ground seeds selection.
2. Fitted surface built based on least-squares fitting.
3. Revised elevations calculation.

Step iii: Ground point extraction using EM
1. Apply EM algorithm to the revised elevations.
2. Calculate the posterior probability $P(G|p)$.
3. If $P(G|p) > 0.5$, the corresponding point is labeled as ground points.

Because point clouds are generally irregularly distributed, it is inefficient to locate one point's neighbors. In this paper, the neighbors of one point refer to the nearest points of one point in the x and y directions. The brute force nearest neighbor search has computational complexity O(n), which is linear to the number of points n. To speed up the search, this paper organizes point clouds using a *k*-dimensional (*k*d) tree [41]. The *k*d-tree algorithm splits the data in half at each level of the tree on the dimension for which the data exhibit the greatest variance. The recursive binary splitting can reduce the average complexity to O(log n). Each non-leaf node (referring to each LiDAR point in the subspace) can be seen as a segmentation hyper plane, which is perpendicular to the coordinate axis and divides the space into two parts. The main steps of the *k*d-tree construction are as follows:

i. Determine the split domain. Calculate the variance of the LiDAR data in the x and y dimensions. Choose the dimension corresponding to the larger variance as the split domain. For instance, we choose the x dimension as the split domain.
ii. Find the segmentation hyper plane. Sort the LiDAR data in the x direction and find the midpoint as the segmentation node. The hyper plane is the one through the segmentation node and perpendicular to the x-axis.
iii. Determine the left subspace and right subspace. According to the hyper plane, the points with x coordinates less than the segmentation node belong to the left subspace, while the points with x coordinates greater than the segmentation node belong to the right subspace.

Apply these three steps to the left and right subspaces. The *k*d-tree construction is a recursive process. The iteration ends when there is only one point contained in the subspace. When the *k*d-tree is built, it will be easy to find one LiDAR point's *k* nearest neighbors. Because the elevations of the outliers are distinctly different from their neighbors, for instance, extremely high or low, the *k* value has little effect on the final denoising results. *k* can be set as a constant, such as 5, 7, or 9. In general, *k* is set to 10, as suggested by Weinmann et al. (2015) [42].

A point is eliminated if its elevation value changes greatly before and after the morphological opening operation among its *k* neighbors. The morphological opening operation is achieved by performing an erosion of the dataset followed by a dilation given as Eq. (1):

$$\begin{cases} E(p) = \min\limits_{(X_i,Y_i)\in Neighbors}(Z_i) \\ D(p) = \max\limits_{(X_i,Y_i)\in Neighbors}(Z_i) \\ O(p) = D(E(p)) \end{cases} \quad (1)$$

where (X_i, Y_i, Z_i) are the coordinates of the neighbors of a point p. E, D and O are the morphological erosion, dilation and opening operations, respectively. As shown in Fig. 2(a), there are three kinds of LiDAR points, including ground points, non-ground points and outliers. Morphological erosion selects the lowest elevation as the point's new elevation among its neighbors. Thus, the elevations of some non-ground points, ground points and outliers are lowered to a similar elevation as their neighbors, as shown in Fig. 2(b). The morphological dilation selects the highest elevation as the point's new elevation among its

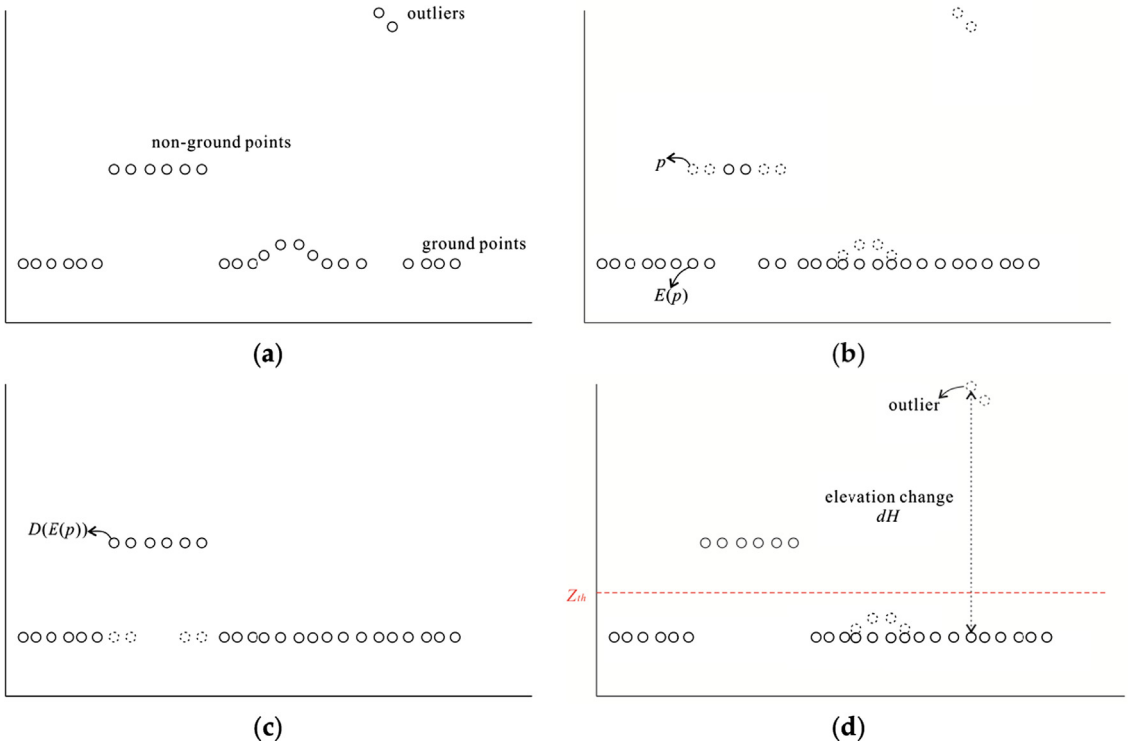

Fig. 2. Diagrammatic sketch of the morphological operations: (a) LiDAR points with outliers; (b) morphological erosion result; (c) result of morphological dilation after erosion; (d) outlier detection based on elevation changes.

neighbors. Note that the morphological opening applies the dilation to the erosion result. Thus, it can be found in Fig. 2(c) that some non-ground points' elevations are recovered after the dilation because their neighbors own higher elevations. According to the morphological opening result, the elevation change dH for each point can be calculated as Eq. (2).

$$dH = Z_p - O(p) \tag{2}$$

where Z_p is the point p's elevation, and $O(p)$ is the point p's new elevation after morphological opening. It is easy to find that the elevation changes for outliers (Fig. 2(d)) are obviously large. Therefore, the outliers can be detected if the elevation changes are greater than the threshold as shown in Fig. 2(d). This can be represented in Eq. (3) as

$$\begin{cases} abs(dH) > Z_{th},\ p \in outliers \\ abs(dH) \leqslant Z_{th},\ p \notin outliers \end{cases} \tag{3}$$

The threshold for each point can be calculated automatically according to Eq. (4):

$$\begin{cases} Z_{th} = 3 * Z_{std} \\ Z_{std} = \sqrt{\frac{1}{k}\sum_{m=1}^{k}(Z_m - Z_{mean})^2} \\ Z_{mean} = \frac{1}{k}\sum_{m=1}^{k} Z_m \end{cases} \tag{4}$$

where Z_{th} is the threshold for detecting outliers, Z_{std} is the elevation standard deviation of the neighbors, Z_{mean} is the mean elevation of the neighbors, and Z_m is the elevation of the m-th point among the point p's k neighbors. Obviously, the threshold for detecting outliers is different from point to point, since their neighbors are generally different.

2.2. Revised elevation calculation

As shown in Fig. 3, when encountering abrupt terrains, non-ground points are commonly mixed with ground points. Some ground point elevations are even higher than those of the non-ground points. For instance, the elevation of the ground point P_1 on the abrupt terrain is obviously larger than that of the non-ground point P_2 on the building roof. Thus, if we directly apply the EM algorithm to separate the ground points and non-ground points, some ground points on the abrupt terrains, such as P_1, will be misclassified as non-ground points, while some non-ground points with lower elevations, such as P_2, will be wrongly classified as ground points. As a result, the method using the EM algorithm cannot perform well in abrupt terrain environments.

The key to correctly separating point clouds using the EM algorithm is that the non-ground points should be located higher than the ground points. Thus, instead of using elevations of LiDAR points, this paper applies the EM algorithm to the revised elevations, which can be calculated by subtracting the interpolated elevations provided by a fitted surface. This can be expressed using Eq. (5).

$$Z_{re} = Z - Z_{\text{int}} \tag{5}$$

where Z is the original elevation, Z_{int} is the interpolated elevation, and Z_{re} is the revised elevation.

In Fig. 3, the dotted line represents the fitted surface built by the ground seeds, which will be described in Sections 2.2.1 and 2.2.2. The fitted surface can be seen as a rough DTM. According to the fitted surface, the revised elevation for each LiDAR point can be obtained. From Fig. 3, it is easy to find that although the elevation of the ground point P_1 is obviously higher than that of the non-ground point P_2, the revised elevation h_1 turns out to be lower than h_2. Thus, most revised elevations of non-ground points are elevated, while the revised elevations of ground points are changed to be smaller. Therefore, on the basis of the revised elevations, the mixed ground points and non-ground points can be distinguished effectively using the EM algorithm.

2.2.1. Ground seed selection

To build a fitted surface, one important step is ground seeds selection. The ground seeds are points with the lowest elevations in the local areas. By this means, we can guarantee that the built surface reflects the

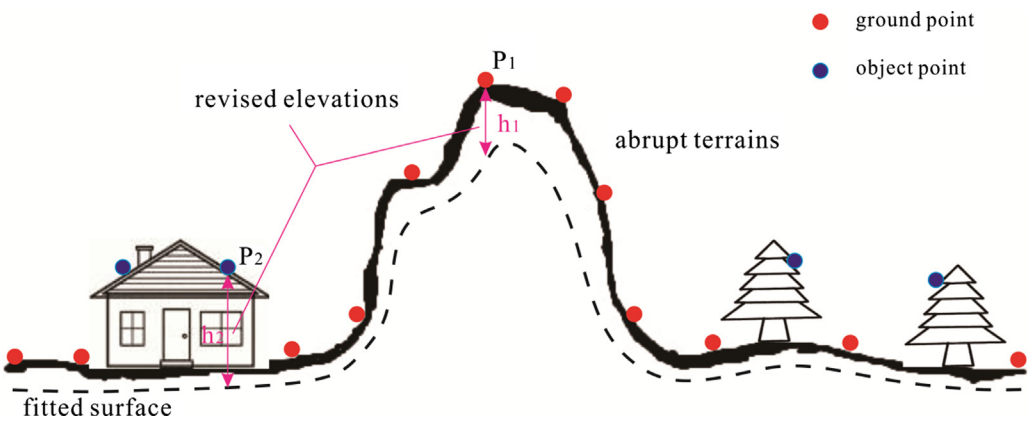

Fig. 3. Revised elevation calculation based on a fitted surface.

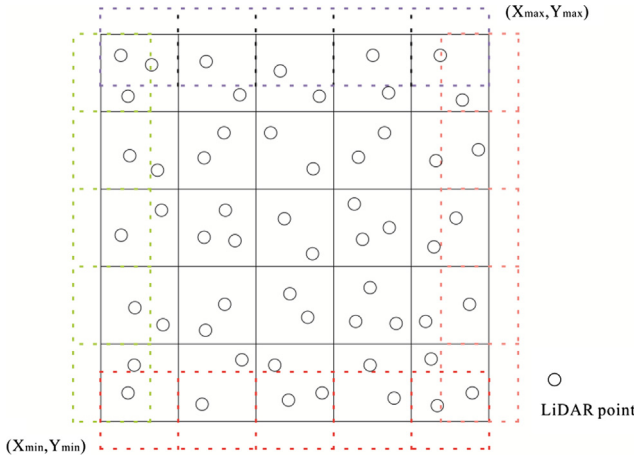

Fig. 4. Grids moved in four directions. In some cells, there is more than one LiDAR point, and only the lowest point is retained as a ground seed.

actual terrain fluctuation. To obtain the lowest points, the LiDAR points are organized as grid cells. As shown in Fig. 4, although more than one point may fall into one cell, only the lowest point is retained as a ground seed. All the lowest points in these cells are used for the following fitted surface generation (Section 2.2.2). The grid cells are built according to Eqs. (6) and (7).

$$\begin{cases} ID_i = floor\left(\frac{X_p - X_{\min}}{cellsize}\right) + 1 \\ ID_j = floor\left(\frac{Y_p - Y_{\min}}{cellsize}\right) + 1 \end{cases} \tag{6}$$

$$\begin{cases} M = floor\left(\frac{X_{\max} - X_{\min}}{cellsize}\right) + 1 \\ N = floor\left(\frac{Y_{\max} - Y_{\min}}{cellsize}\right) + 1 \end{cases} \tag{7}$$

where $X_{\min}$ and $Y_{\min}$ are the minimum coordinates of the point clouds, while $X_{\max}$ and $Y_{\max}$ are the maximum coordinates of the point clouds. *cellsize* is the size of the grid, which is defined as a constant. It should be larger than the maximum object (mainly refers to the building roof) size. Thus, 50–70 m is generally suitable [36]. $floor(\Delta)$ is used to return the largest integral value that is not larger than Δ. ID_i and ID_j are the grid indexes of the point p. M and N are the total number of grids in the horizontal direction and vertical direction, respectively.

To ensure that the ground seeds are not non-ground points, the grid size is generally larger than the largest objects in the test site. Therefore, the number of ground seeds may not be enough to build an accurate fitted surface. To obtain more ground seeds, this paper makes the pseudo grids move gradually, as shown in Fig. 4. The grids are moved in four directions up and down. The moving step is equal to one-third of the width of a grid.

2.2.2. Fitted surface built based on least-squares fitting

This paper adopts the Least-Squares Fitting method to build the fitted surface, which is defined as a quadratic polynomial formula (Eq. (8)):

$$f(X, Y) = \lambda_0 + \lambda_1 X + \lambda_2 Y + \lambda_3 XY + \lambda_4 X^2 + \lambda_5 Y^2 \tag{8}$$

where λ_0, λ_1, λ_2, λ_3, λ_4 and λ_5 are coefficients that can be solved using at least six ground seeds. The error v_i of each ground seed can be calculated according to Equation (9).

$$v_i = \lambda_0 + \lambda_1 X_i + \lambda_2 Y_i + \lambda_3 X_i Y_i + \lambda_4 X_i^2 + \lambda_5 Y_i^2 - Z_i \tag{9}$$

where (X_i, Y_i, Z_i) is the coordinate of the ground seed. Using all the ground seeds selected in Section 2.2.1, the coefficients λ can be calculated according to the least square principle. Finally, the fitted surface can be built based on these coefficients. Fig. 5(a) shows a digital surface model (DSM) with complex terrain features. According to the principle presented in Section 2.2.1, grid cells can be constructed as shown in Fig. 5(b). Obviously, many LiDAR points will fall into one cell. To acquire ground seeds, only the point with the lowest elevation is retained. Finally, all ground seeds are used to build a fitted surface according to the Least-Squares Fitting method, as shown in Fig. 5(c). Note that the quadratic polynomial is fitted for the whole area using all the ground seeds.

2.3. Ground points extraction using EM

To realize the filtering, the posterior probability of a point p_i ($i = 1, 2, \ldots, n$, n is the total number of the LiDAR points) belonging to ground points (G) should be calculated. It can be determined according to Eq. (10):

$$P(G\,|p_i) = \frac{P(p_i\,|G)P(G)}{P(p_i)} \tag{10}$$

Similarly, the posterior probability of the point p_i belonging to non-ground points (NG) can also be calculated according to Eq. (11).

$$P(NG\,|p_i) = \frac{P(p_i\,|NG)P(NG)}{P(p_i)} \tag{11}$$

where $P(p_i)$ is equal to $P(p_i\,|G)P(G) + P(p_i\,|NG)P(NG)$, $P(G)$ and $P(NG)$ are the prior probability of ground points and non-ground points, respectively.

It is obvious that p_i will be labeled as a ground point if $P(G\,|p_i)$ is greater than $P(NG\,|p_i)$. Since there are two classes, namely, ground points and non-ground points, formed the mixed Gaussian models, $P(G)$ and $P(NG)$ are equal to 0.5. Thus, to obtain the posterior probability of

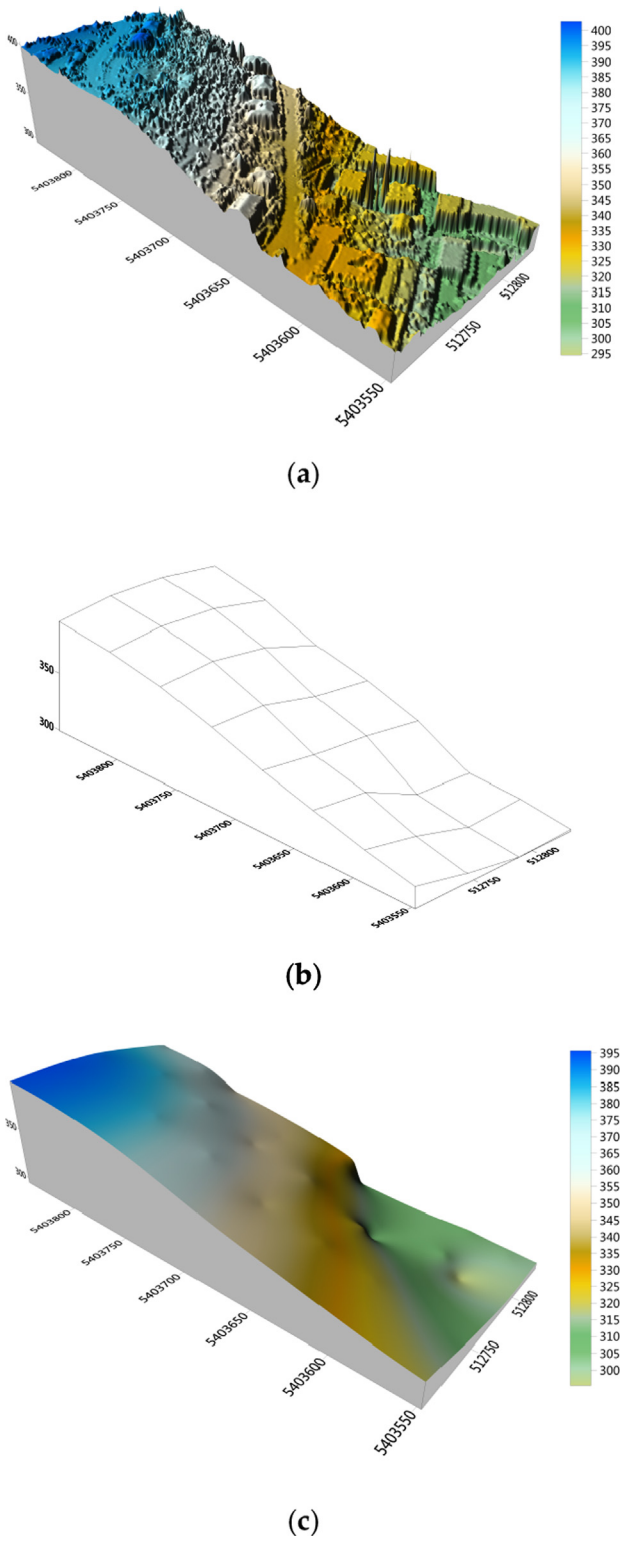

Fig. 5. Example of the fitted surface building based on ground seeds: (a) DSM with complex terrain features; (b) grid cell construction for selecting ground seeds; (c) fitted surface building using the ground seeds.

a point belonging to ground points ($P(G\ |p_i)$), we need to calculate the class-conditional density $P(p_i\ |G)$ given in Eq. (12), which can be estimated using the EM algorithm. Similarly, $P(p_i\ |NG)$ can also be estimated according to Eq. (12) except for the parameters (C, μ and δ) values are different from the ones of $P(p_i\ |G)$.

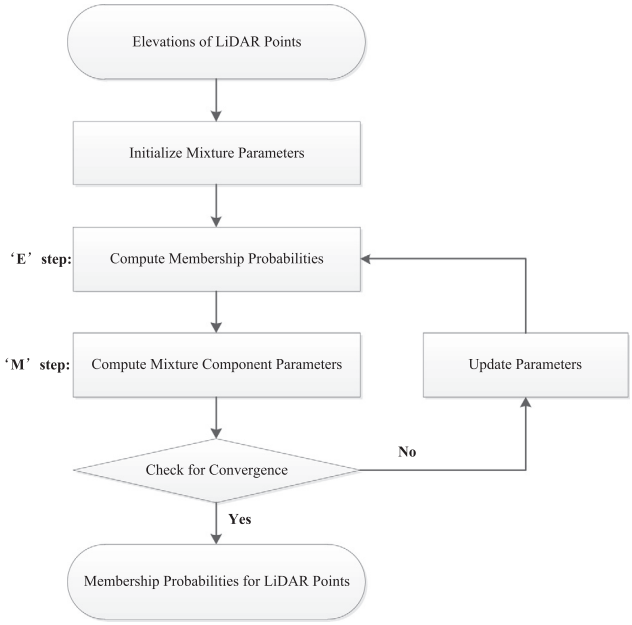

Fig. 6. Flow chart of the EM algorithm.

Table 2
Features contained in the 15 samples.

Region	Point Spacing	Samples	Features
Urban	1.0–1.5 m	samp11 samp12 samp21 samp22 samp23 samp24	Steep slopes, dense vegetation, data gaps, and buildings on hillsides Irregular buildings, road networks, tunnels, bridges, and road networks
		samp31	Dense buildings surrounding by vegetation, data gaps, and mixture of high and low objects
		samp41 samp42	Railway station with trains, data gaps, and dense points
Rural	2.0–3.5 m	samp51 samp52 samp53 samp54	Steep slopes with vegetation, data gaps, and terrain discontinuity
		samp61	Buildings, embankments, data gaps, and roads
		samp71	Bridges, embankments, underpasses, and roads

Table 3
Metrics of the three accuracy indices. a and b are the numbers of ground points classified correctly and incorrectly; d and c are the numbers of non-ground points classified correctly and incorrectly.

		Filtering results		Metrics of Evaluation
		Ground	Non-ground	$O\_error = b/(a + b)$
Reference	Ground	a	b	$C\_error = c/(c + d)$
	Non-ground	c	d	$Total = (b + c)/(a + b + c + d)$

$$P(p_i\ |G) = \sum_{j=1}^{2} C_j Gaussian(Z_i\ |\mu_j, \delta_j) \tag{12}$$

where C_j is the mixing coefficient, Z_i is the elevation of the point p_i, $Gaussian(\cdot)$ is the Gaussian equation with parameters μ_j and δ_j, which are the mean and standard deviation of elevations. It can be denoted as Eq. (13).

Table 4
Three types of error comparison among the proposed EM algorithm, the PTD algorithm, and the Segment-based PTD algorithm (the bold value is the smallest one).

ISPRS Samples	Filtering algorithms								
	The proposed algorithm			PTD algorithm			Segment-based PTD algorithm		
	Omission %	Commission %	Total %	Omission %	Commission %	Total %	Omission %	Commission %	Total %
samp11	**19.94**	13.30	**17.10**	46.68	**3.40**	28.21	25.67	8.84	18.49
samp12	8.55	5.68	7.14	15.60	**1.92**	8.93	**8.13**	3.61	**5.92**
samp21	1.79	**5.23**	**2.55**	**0.78**	10.47	2.93	1.17	18.23	4.95
samp22	**2.90**	30.45	**11.47**	36.84	**3.23**	26.36	19.05	3.44	14.18
samp23	**2.23**	16.27	**8.86**	35.33	**3.82**	20.42	19.25	4.05	12.06
samp24	**19.44**	**6.80**	**15.96**	40.30	12.54	32.67	22.86	13.41	20.26
samp31	**0.21**	14.57	6.82	3.93	3.55	3.76	2.10	**2.59**	**2.32**
samp41	**12.28**	10.63	**11.45**	60.34	**0.91**	30.55	39.54	1.44	20.44
samp42	11.65	**1.02**	4.13	12.13	1.45	4.58	**9.72**	1.55	**3.94**
samp51	5.56	7.22	5.92	4.91	**3.80**	**4.67**	**2.05**	16.97	5.31
samp52	**12.05**	8.64	**11.69**	19.20	**4.95**	17.70	12.53	16.77	12.98
samp53	26.22	20.52	25.98	26.66	**1.44**	25.64	**4.25**	37.22	**5.58**
samp54	8.46	3.07	5.56	8.76	**2.53**	**5.41**	**3.59**	8.82	6.40
samp61	26.56	7.97	25.92	18.52	2.82	17.98	**16.62**	**2.49**	**16.13**
samp71	**5.84**	9.94	**6.30**	16.81	**3.50**	15.30	10.07	13.39	10.44
Ave	10.91	10.75	11.12	23.12	4.02	16.34	13.11	10.19	10.63
Min	0.21	1.02	2.55	0.78	0.91	2.93	1.17	1.44	2.32
Max	26.56	30.45	25.98	60.34	12.54	32.67	39.54	37.22	20.44
Std	8.33	7.24	7.04	16.82	3.14	10.35	10.45	9.32	6.01

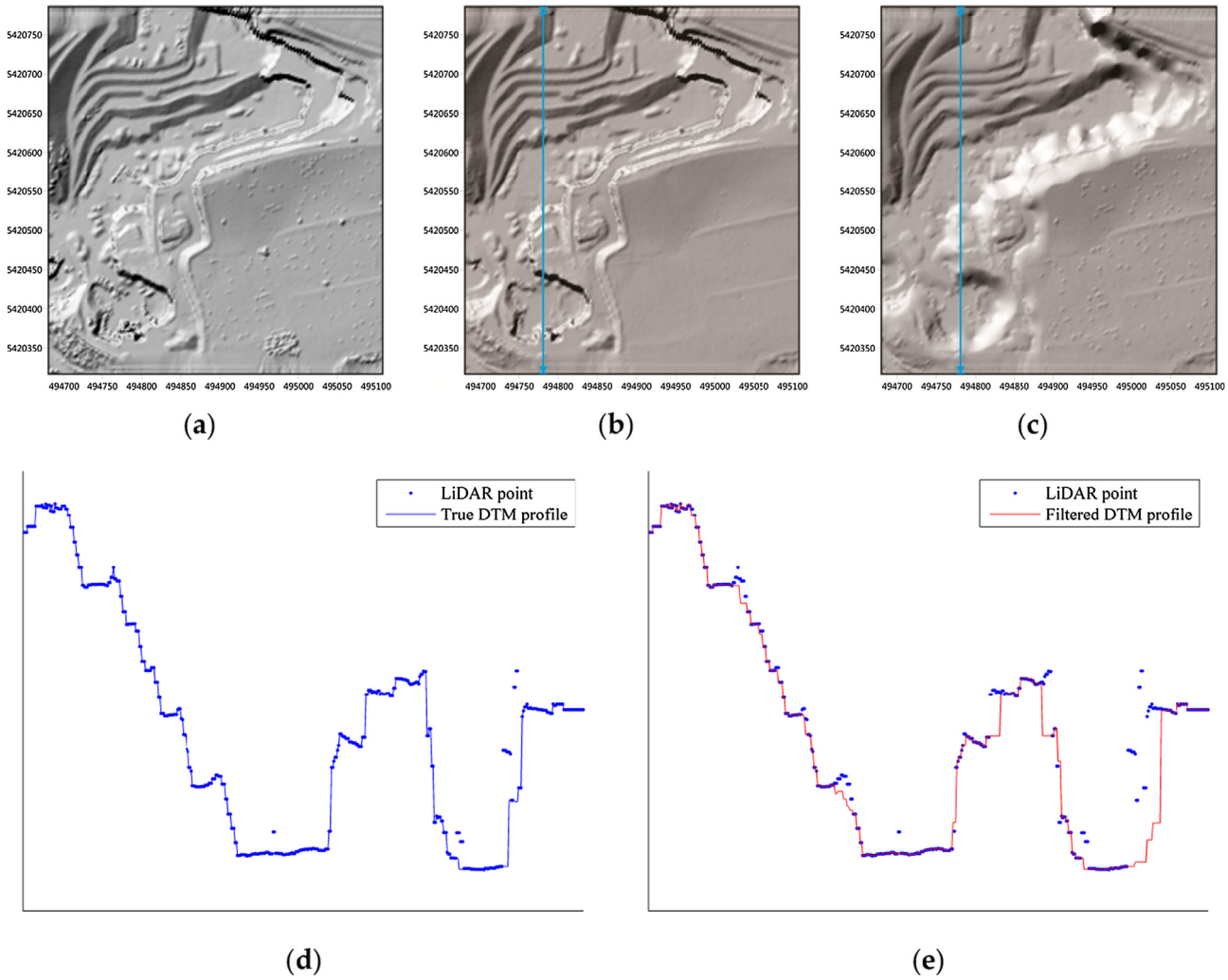

Fig. 7. Filtering result of samp53: (a) the DSM before filtering; (b) the true DTM generated from the true ground points; (c) the filtered DTM generated from the ground points derived by the proposed filtering algorithm; (d) the cross-section profile (blue line in (b)) of the true DTM; (e) the cross-section profile (blue line in (c)) of the filtered DTM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

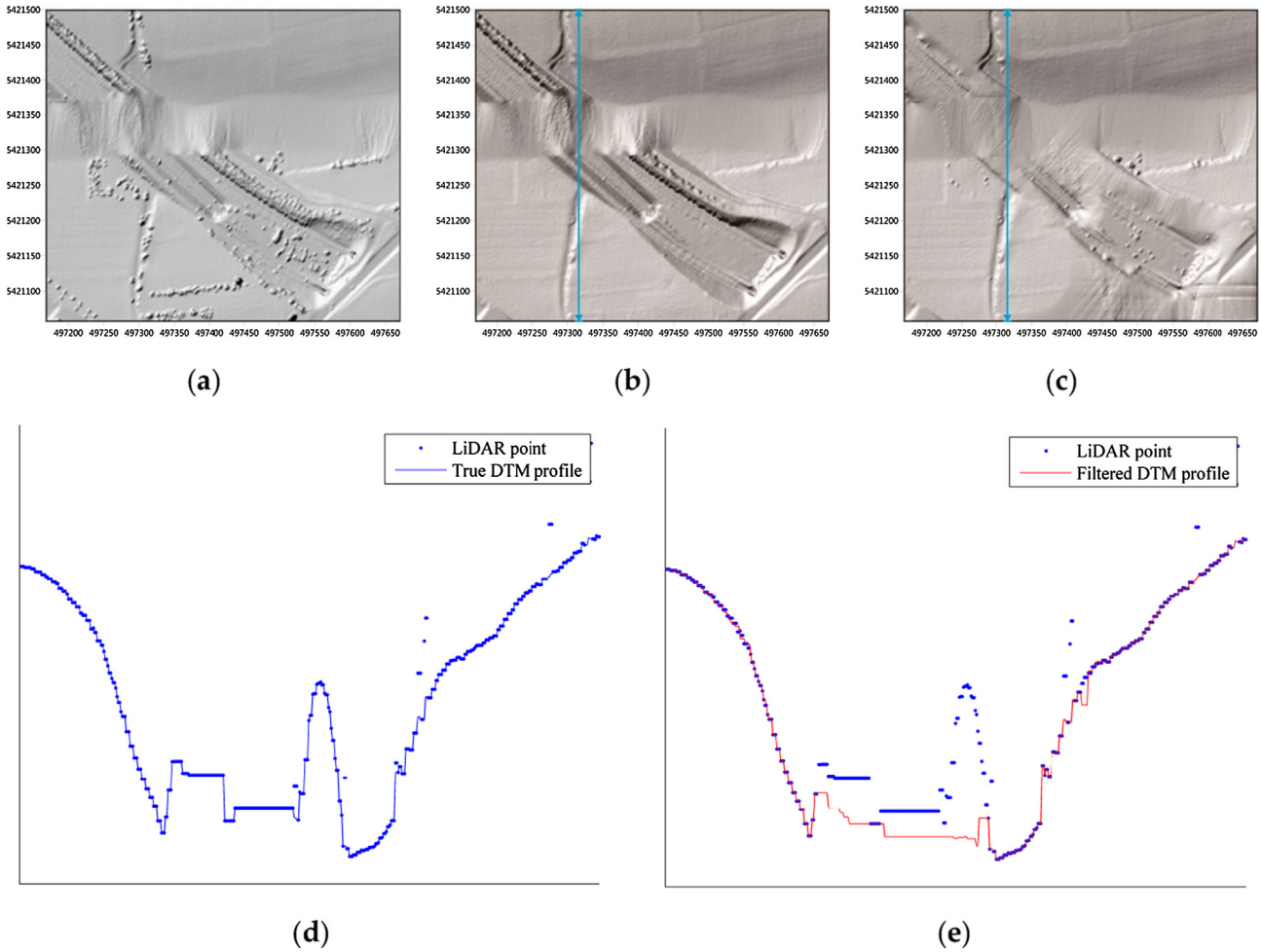

Fig. 8. Filtering result of samp61: (a) the DSM before filtering; (b) the true DTM generated from the true ground points; (c) the filtered DTM generated from the ground points derived by the proposed filtering algorithm. (d) The cross-section profile (blue line in (b)) of the true DTM; (e) the cross-section profile (blue line in (c)) of the filtered DTM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$Gaussian(Z_i \mid \mu_j, \delta_j) = \frac{1}{\sqrt{2\pi}\delta_j} \times e^{-\frac{(Z_i-\mu_j)^2}{2\delta_j^2}} \tag{13}$$

The EM algorithm is a general method for fitting probability distributions. The flow chart of this algorithm is shown in Fig. 6. It includes the following four steps [43].

i. Initialize mixture parameters, including C_j, μ_j and δ_j, $j = 1, 2$.

In Eq. (12), three kinds of parameters (C, μ and δ) need to be initialized. Since there are two components that form the mixed Gaussian model, C_1 and C_2 can be initialized as 0.5. Of course, some other values are also effective only if C_1 plus C_2 is equal to one. μ_1, μ_2, δ_1 and δ_2 can be initialized according to Eqs. (14) and (15).

$$\begin{cases} m_0 = \frac{1}{n}\sum_{i=1}^{n} Z_i \\ sigma_0 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_i - m_0)^2} \end{cases} \tag{14}$$

$$\begin{cases} \mu_1 = \mu_2 = m_0 + sigma_0 \times randn(1) \\ \delta_1 = \delta_2 = sigma_0 \end{cases} \tag{15}$$

where Z_i is the elevation of the point, n is the number of LiDAR points, and $randn(1)$ generates one random normal distribution value.

ii. 'E' step: compute membership probabilities, namely, $P(G \mid p_i)$ and $P(NG \mid p_i)$.

The membership probabilities (belonging to ground points or non-ground points) for each point can be calculated according to Eqs. (10)–(13).

iii. 'M' step: update mixture component parameters

These parameters are C_j, μ_j and δ_j, $j = 1, 2$. They can be calculated according to Eqs. (16)–(18).

$$\begin{cases} C_1 = \sum_{i=1}^{n} P(G \mid p_i) \Big/ \left(\sum_{i=1}^{n} P(G \mid p_i) + \sum_{i=1}^{n} P(NG \mid p_i) \right) \\ C_2 = \sum_{i=1}^{n} P(NG \mid p_i) \Big/ \left(\sum_{i=1}^{n} P(G \mid p_i) + \sum_{i=1}^{n} P(NG \mid p_i) \right) \end{cases} \tag{16}$$

$$\begin{cases} \mu_1 = \sum_{i=1}^{n} \left(P(G \mid p_i) \times Z_i \Big/ \sum_{i=1}^{n} P(G \mid p_i) \right) \\ \mu_2 = \sum_{i=1}^{n} \left(P(NG \mid p_i) \times Z_i \Big/ \sum_{i=1}^{n} P(NG \mid p_i) \right) \end{cases} \tag{17}$$

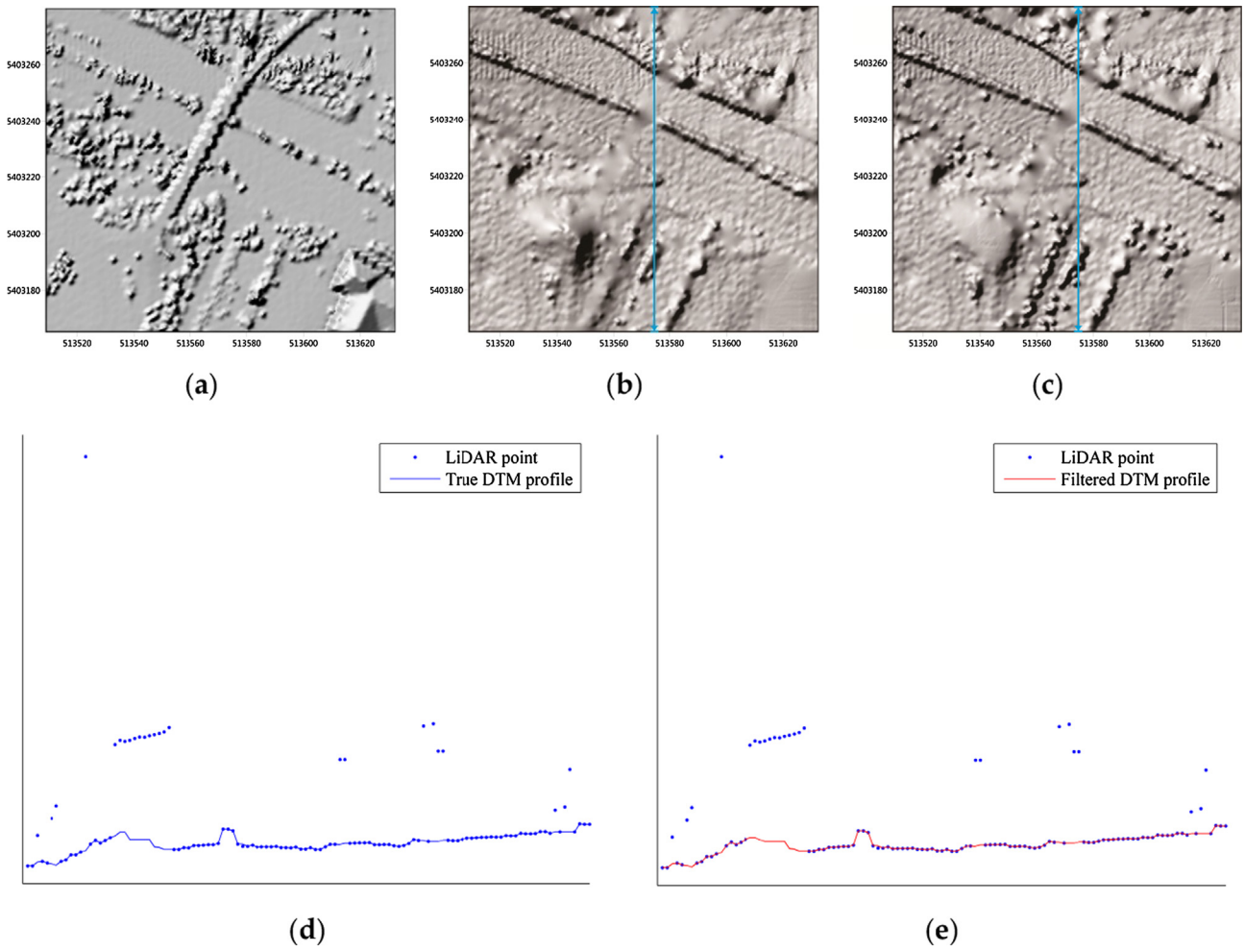

Fig. 9. Filtering result of samp21: (a) the DSM before filtering; (b) the true DTM generated from the true ground points; (c) the filtered DTM generated from the ground points derived by the proposed filtering algorithm. (d) The cross-section profile (blue line in (b)) of the true DTM; (e) the cross-section profile (blue line in (c)) of the filtered DTM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\begin{cases} \delta_1 = \sqrt{\sum_{i=1}^{n} ((Z_i - \mu_1)^2 \times P(G \mid p_i)) \Big/ \sum_{i=1}^{n} P(G \mid p_i)} \\ \delta_2 = \sqrt{\sum_{i=1}^{n} ((Z_i - \mu_2)^2 \times P(NG \mid p_i)) \Big/ \sum_{i=1}^{n} P(NG \mid p_i)} \end{cases} \quad (18)$$

where Z_i is the elevation of the point p_i, n is the total number of LiDAR points.

iv. Check for convergence

The EM algorithm continues iterating steps ii-iv until it reaches convergence. The convergence can be estimated by the following conditions:

Providing that C_j^{old}, μ_j^{old} and δ_j^{old} $(j = 1, 2)$ are the parameters of the last iteration, while C_j^{new}, μ_j^{new} and δ_j^{new} $(j = 1, 2)$ are the newly calculated parameters. The iteration ends when the absolute norm changes of the three kinds of parameters are all smaller than the tolerance. The tolerance can be calculated according to Eq. (19).

$$tol = sigma_0 \times 10^{-8} \quad (19)$$

where $sigma_0$ is the standard deviation of the LiDAR points' elevations, which can be calculated according to Eq. (14).

3. Experimental results and analysis

3.1. Test datasets

The test data are the standard filtering benchmark datasets provided by the ISPRS Commission III/WG3 (http://www.itc.nl/isprswgIII-3/filtertest/). The datasets located in the Vaihingen/Enz test field and Stuttgart city center were collected using an Optech ALTM scanner. There are 15 samples in the datasets, which cover different challenging filtering features as described in Table 2. The 15 samples consist of nine urban areas (samp11-samp42) with point spacing of 1–1.5 m and six rural areas (samp51-samp71) with point spacing of 2–2.5 m. Moreover, for each sample, the reference data are manually selected by the ISPRS for quantitative assessment of the filtering performance.

3.2. Accuracy indices

Three accuracy indices namely omission error (*O_error*), commission error (*C_error*) and total error (*Total*) were adopted in this study to assess the filtering performance of the proposed EM method. Omission error is the percentage of rejected ground points as non-ground points, while commission error denotes the percentage of accepted non-ground points as ground points. Total error is the percentage of all misclassified points. The metrics of the three accuracy indices are shown in Table 3.

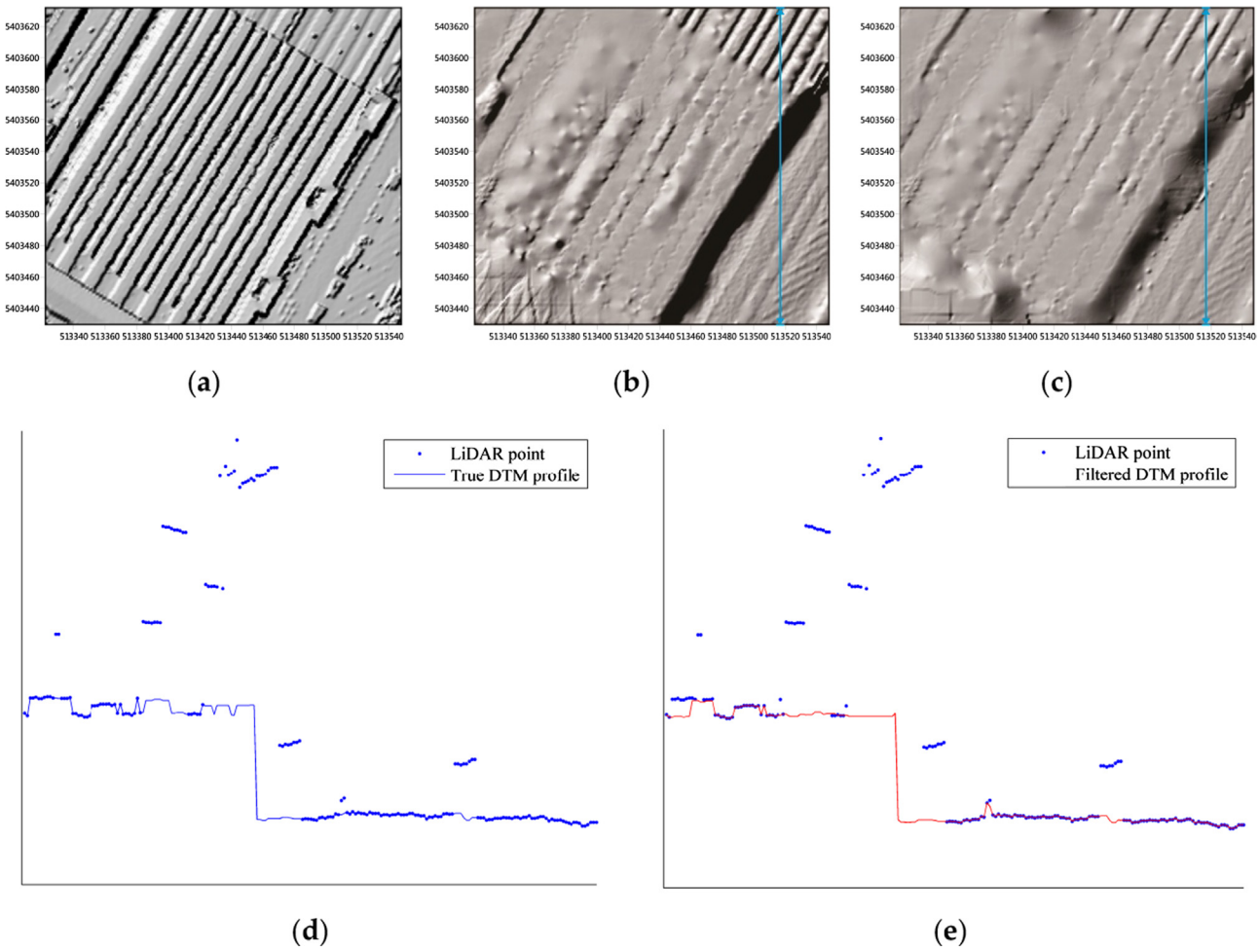

Fig. 10. Filtering result of samp42: (a) the DSM before filtering; (b) the true DTM generated from the true ground points; (c) the filtered DTM generated from the ground points derived by the proposed filtering algorithm. (d) The cross-section profile (blue line in (b)) of the true DTM; (e) the cross-section profile (blue line in (c)) of the filtered DTM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.3. Comparison and assessment

To quantitatively evaluate the performance of the proposed method, this paper calculated the abovementioned three types of errors (Table 3) and compared them with those of the classic PTD method [29] and an improved PTD method based on segmentation proposed recently [34]. PTD has been a popular algorithm, not only because of its effectiveness and robustness in filtering, but also because of its application in the TerraScan commercial software. However, the PTD method may fail to preserve terrain details in areas with steep slopes [34]. To solve this problem, a segment-based PTD method is proposed. The main improvement of this method is embedding segmentation to expend as many initial ground points as possible. The segment-based PTD method also tests on the 15 samples provided by the ISPRS. Notably, both the PTD and the segment-based PTD methods require fine tuning of the threshold parameters, such as iterative angle and distance, which directly affect the filtering results. This paper also analyzed the average value, minimum value, maximum value and standard deviation of these three errors of the three methods (EM, PTD and segment-based PTD). The comparison results are shown in Table 4.

From Table 4, it can be observed that the proposed method obtained lower omission error for 8 samples, including samp11, samp22, samp23, samp24, samp31, samp41, samp52, and samp71. As a result, the average omission error of the proposed method is the lowest in comparison with the PTD and the segment-based PTD methods. The average omission error of the proposed method is 52.81% and 16.78% lower than the classic PTD method and the segment-based PTD method, respectively. In terms of the commission error, the proposed method and the segment-based PTD method achieve similar accuracy. Both of their average commission errors are close; however, they are much higher than that of the PTD method. Although the commission error is higher for most samples, the proposed method can reduce its average total error by 31.95% compared to the PTD method. This would significantly decrease the cost of manual operation required in post-processing. When comparing the average errors, it can be found that the proposed method achieves a good balance between the average omission error (10.91%) and the commission error (10.75%). This means that the proposed method can detect the non-ground points effectively while preserving terrain details as much as possible. In terms of standard deviation, the proposed method also performs much better than the PTD method. All the standard deviations of the three types of errors are relatively low, which indicates that the proposed method is robust enough to achieve good filtering performance in different terrain environments.

It can also be observed that the proposed method obtained the largest total errors for samp53 and samp61, while samp21 and samp42 achieved the smallest total errors. This paper further analyzed the four areas before and after filtering as shown in Figs. 7–10. Fig. 7(a) shows that there are many terrain discontinuities in the area of samp53. Although this paper built a fitted surface based on Least-Squares to revise the elevations of each point, many ground points are still rejected as non-ground points (Fig. 7(c)) to lead to the omission error being larger.

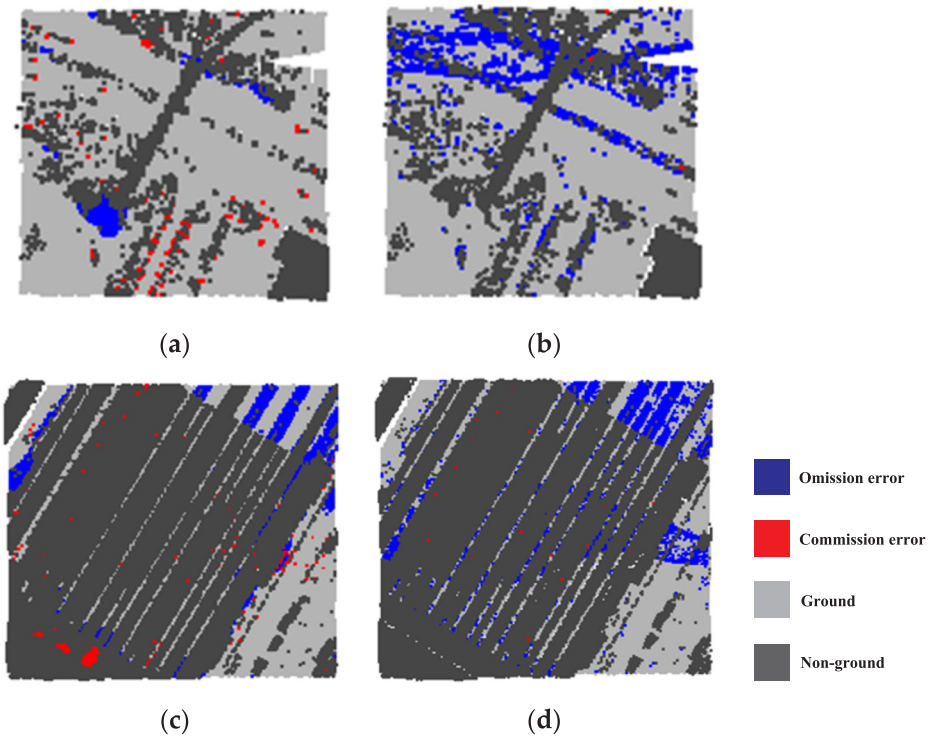

Fig. 11. The differences between the truth and the estimated values. (a) The filtering result of the proposed method for samp21; (b) the filtering result of the PTD method for samp21; (c) the filtering result of the proposed method for samp42; (d) the filtering result of the PTD method for samp42.

Compared with the cross-section profiles of the true DTM and the filtered DTM (Fig. 7(d) and (e)), it can also be found that some ground points are wrongly classified. The reason that the commission error is higher can be explained by the fact that there are few non-ground points in this area and even a small number of misclassified points can make commission error become larger. As shown in Fig. 8(a), there are embankments in the test area of samp61. The elevations of embankments are similar to those of terrains. In addition, the slope of this area is slightly larger. Thus, the filtering result in this area is prone to error. Fortunately, most non-ground points are eliminated effectively when comparing Fig. 8(b) and (c). However, some terrain details are flattened as shown in Fig. 8(d) and (e).

The greatest filtering challenge of samp21 is the attached object, namely, bridge. Compared to the reference filtering result (Fig. 9(b)), the proposed EM method removed the bridge effectively, as shown in Fig. 9(c). Because the terrain is relatively flat, the proposed EM method performed best in this area. It can also be confirmed by the comparison of Fig. 9(d) and (e) that almost all the LiDAR points are correctly classified as ground points and non-ground points, respectively. In the test area of samp42, a railway station with trains is the main object (Fig. 10(a)). Due to the apparent elevation differences between the terrains and objects, only a few points are misclassified (Fig. 10(b) and (c)). The cross-section profiles as shown in Fig. 10(d) and (e) also prove this. Fig. 11 shows the differences between the truth and the estimated values. It describes the spatial distributions of the omission and commission errors. In comparison with Fig. 11(a) and (b), it can be found that the proposed method achieved much lower omission error than that of the PTD method for samp21. Fig. 11(c) and (d) also show that there are fewer points misclassified in the result of the proposed method. That is why the total error of the proposed method (4.13%) is lower than that of the PTD method (4.58%) for samp42 (Table 4).

Since the ISPRS datasets were obtained almost twenty years ago, this paper further accessed the performance of the proposed method using another dataset used in practice. The dataset was acquired using an Optech ALTM scanner for the city of Jingmen in China with an area of 1.20 km^2 and 984,998 points in total. The average point density of this dataset was 0.82 points/m^2. This dataset covers diverse land-use and land-cover types including residential buildings, roads, forests and farmlands as shown in Fig. 12(a). As reported by Maguya et al. (2014), some challenging conditions with few ground points, such as dense canopy (red rectangle marked areas in Fig. 12(a)), pose greater challenges for the filtering algorithm [44]. From the filtering results shown in Fig. 12(c), the proposed method performs well in these areas. This is because the fitted surface is built using the ground seeds selected from the whole area. Although there are few ground points in some dense forest areas, the fitted surface can still be interpolated correctly using the ground seeds surrounding these areas. Thus, the revised elevations can also be calculated correctly according to the fitted surface. The reference ground points were obtained using Terrascan software combined with manual editing, as shown in Fig. 12(b). Fig. 12(d) and (e) are cross-section profiles of the true DTM and filtered DTM, respectively. This comparison also proves that the proposed method is sound and useful with this new dataset. However, as shown in Fig. 12(c), the rectangle marked area has a large filtering error compared to the reference result in Fig. 12(b). This may be caused by a non-ground point being wrongly selected as a ground seed, which will lead to the revised elevations of points in this area being wrongly calculated. Obviously, the proposed algorithm cannot perform well using revised elevations with errors. The omission, commission and total errors are 2.81%, 3.45%, and 3.23%, respectively. It can be concluded that the proposed method can achieve good filtering performance on the modern airborne LiDAR dataset even if the dataset contains a larger number of points and covers a larger area.

4. Conclusion

Point cloud filtering is a necessary step in point cloud processing, analysis and applications. To break through the limitation of complex parameter settings for the existing filtering algorithms, this paper proposed a threshold-free filtering algorithm based on expectation-

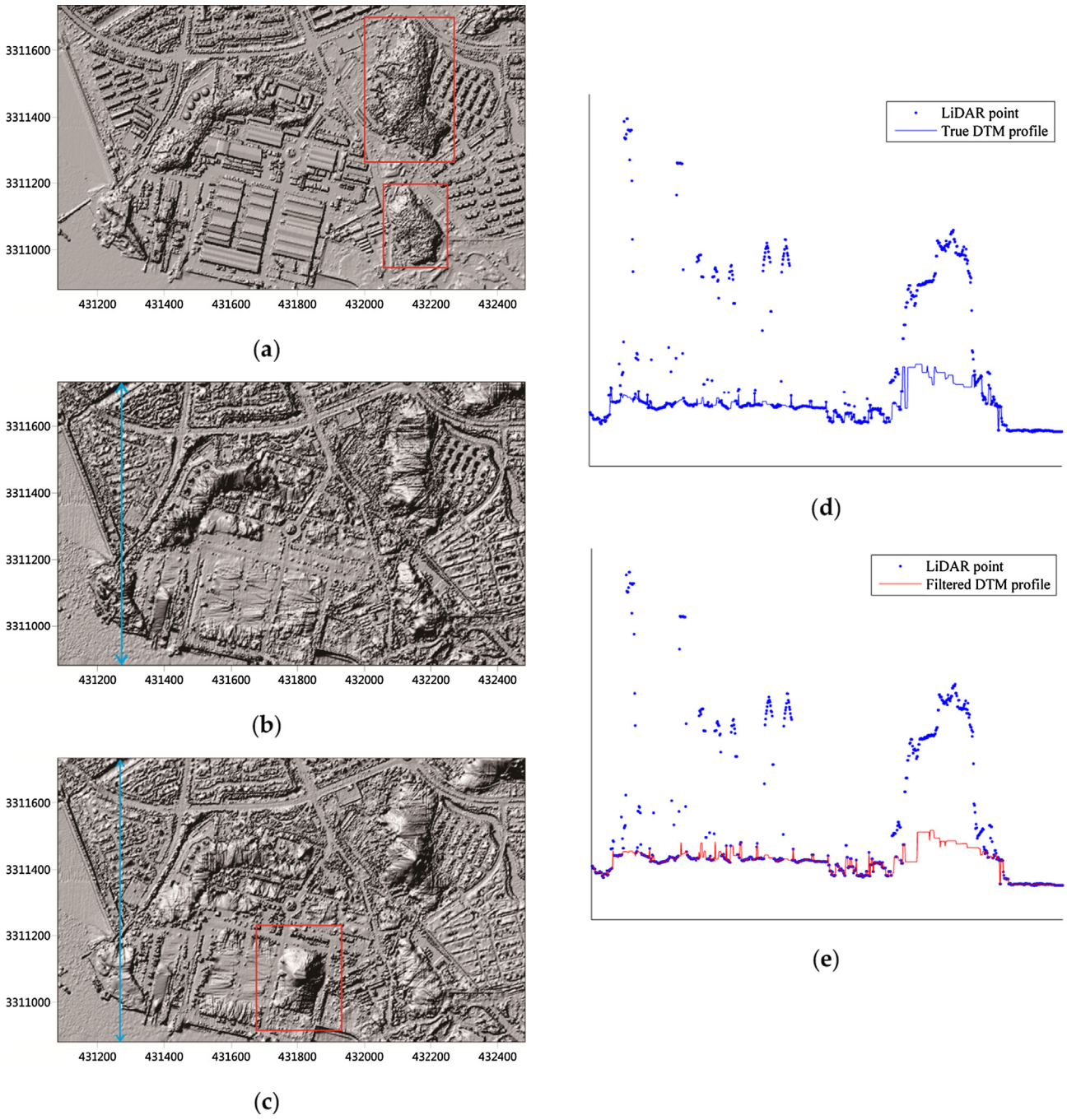

Fig. 12. Filtering result of Jingmen: (a) the DSM before filtering; (b) the true DTM generated from the true ground points; (c) the filtered DTM generated from the ground points derived by the proposed filtering algorithm. (d) The cross-section profile (blue line in (b)) of the true DTM; (e) the cross-section profile (blue line in (c)) of the filtered DTM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

maximization. In this paper, filtering is seen as a separation of mixed Gaussian models. By applying the EM algorithm to revised elevations of point clouds, ground points can be extracted automatically. The revised elevations for each point are calculated based on the fitted surface built by ground seeds using the Least-Squares principle. Experimental results show that the proposed method can achieve better performance without complex parameter setting or threshold adjustments compared with the classic PTD method and the segment-based PTD methods in terms of omission error. The average omission error of the proposed method is 52.81% and 16.78% lower than the classic PTD method and the segment-based PTD method, respectively. Moreover, the proposed method can reduce its average total error by 31.95% compared to the classic PTD method. Furthermore, all the standard deviations of the three types of errors of the proposed method are relatively lower, which indicates that the proposed method is robust enough to achieve good filtering performance in different terrain environments. Further testing carried out on a new dataset located in Jingmen, China confirmed this conclusion. Additionally, the filtering performance is relatively unaffected by the format and resolution of the airborne LiDAR data. Therefore, the proposed method can provide a good foundation for the post-processing of airborne LiDAR point clouds. However, although omission errors of the new method are small, commission errors are slightly larger. Controlling the increment of the commission error to be consistent with the omission error will be the focus in future research.

Acknowledgments

The authors would like to thank the Fund from National Science Foundation (NSF), China (41801325, 41861052, 41874001), Education Department of Jiangxi Province, China (GJJ170449), Key Laboratory for Digital Land and Resources of Jiangxi Province, East China University of Technology, China (DLLJ201806) and East China University of Technology Ph. D. Project, China (DHBK2017155) for their financial support.

Author contributions

Zhenyang Hui had the original idea for the study and drafted the manuscript. Shuanggen Jin and Yao Ziggah Yevenyo contributed to improvement of the original idea and revision of the manuscript. Dajun Li, Leyang Wang and Youjian Hu performed the experiments and experimental analysis.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] X. Meng, N. Currit, K. Zhao, Ground filtering algorithms for airborne LiDAR data: a review of critical issues, Remote Sens. 2 (3) (2010) 833–860.
[2] X. Liu, Airborne LiDAR for DEM generation: some critical issues, Prog. Phys. Geogr. 32 (1) (2008) 31–49.
[3] G. Vosselman, H.G. Maas, Airborne and Terrestrial Laser Scanning, CRC Press Taylor and Francis Group, Boca Raton, FL, USA, 2010.
[4] A.H. Özcan, C. Ünsalan, LiDAR data filtering and DTM generation using empirical mode decomposition, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (1) (2017) 360–371.
[5] Y. Li, B. Yong, P.V. Oosterom, et al., Airborne LiDAR data filtering based on geodesic transformations of mathematical morphology, Remote Sens. 9 (11) (2017) 1104.
[6] G. Zhou, X. Zhou, Seamless fusion of LiDAR and aerial imagery for building extraction, IEEE Trans. Geosci. Remote Sens. 52 (11) (2014) 7393–7407.
[7] B. Yang, R. Huang, J. Li, et al., Automated reconstruction of building LoDs from airborne LiDAR point clouds using an improved morphological scale space, Remote Sens. 9 (1) (2017) 14.
[8] Z. Hui, Y. Hu, S. Jin, et al., Road centerline extraction from airborne LiDAR point cloud based on hierarchical fusion and optimization, ISPRS J. Photogramm. Remote Sens. 118 (2016) 22–36.
[9] A.A. Matkan, M. Hajeb, S. Sadeghian, Road extraction from lidar data using support vector machine classification, Photogramm. Eng. Remote Sens. 80 (5) (2015) 409–422.
[10] C. Vega, J. Renaud, S. Durrieu, et al., On the interest of penetration depth, canopy area and volume metrics to improve Lidar-based models of forest parameters, Remote Sens. Environ. 175 (2016) 32–42.
[11] J. Hyyppä, M. Holopainen, H.K. Olsson, Laser scanning in forests, Remote Sens. 4 (10) (2012) 2919–2922.
[12] M. Flood, LIDAR activities and research priorities in the commercial sector, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. 34 (3/W4) (2001) 3–7.
[13] Z. Chen, B. Gao, B. Devereux, State-of-the-Art: DTM generation using airborne LIDAR data, Sensors 17 (1) (2017) 150.
[14] G. Sithole, G. Vosselman, Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds, ISPRS J. Photogramm. Remote Sens. 59 (1–2) (2004) 85–101.
[15] G. Vosselman, Slope based filtering of laser altimetry data, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. XXXIII (Pt. B3) (2000) 935–942.
[16] A. Sampath, D.E.M. Urban, Generation from raw lidar data: a labeling algorithm and its performance, Photogramm. Eng. Remote Sens. 71 (2) (2005) 217–226.
[17] G. Sithole, Filtering of laser altimetry data using slope adaptive filter, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. XXXIV (Pt. 3/W4) (2001) 203–210.
[18] J. Susaki, Adaptive slope filtering of airborne LiDAR data in urban areas for digital terrain model (DTM) generation, Remote Sens. 4 (12) (2012) 1804–1819.
[19] Q. Chen, H. Wang, H. Zhang, et al., A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas, Remote Sens. 8 (711) (2016).
[20] X. Lin, J. Zhang, Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments, Remote Sens. 6 (2) (2014) 1294–1326.
[21] K.Q. Zhang, S.C. Chen, D. Whitman, et al., A progressive morphological filter for removing nonground measurements from airborne LIDAR data, IEEE Trans. Geosci. Remote Sens. 41 (4) (2003) 872–882.
[22] Q. Chen, P. Gong, D. Baldocchi, et al., Filtering airborne laser scanning data with morphological methods, Photogramm. Eng. Remote Sens. 73 (2) (2007) 175–185.
[23] Y. Li, H. Wu, H. Xu, et al., A gradient-constrained morphological filtering algorithm for airborne LiDAR, Opt. Laser Technol. 54 (32) (2013) 288–296.
[24] Y. Li, B. Yong, H. Wu, et al., Filtering airborne lidar data by modified white top-hat transform with directional edge constraints, Photogramm. Eng. Remote Sens. 80 (2) (2014) 133–141.
[25] T.J. Pingel, K.C. Clarke, W.A. Mcbride, An improved simple morphological filter for the terrain classification of airborne LIDAR data, ISPRS J. Photogramm. Remote Sens. 77 (2013) 21–30.
[26] D. Mongus, N. Lukač, B. žalik, Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces, ISPRS J. Photogramm. Remote Sens. 93 (2014) 145–156.
[27] Z. Hui, Y. Hu, Y.Z. Yevenyo, et al., An Improved Morphological Algorithm for Filtering Airborne LiDAR Point Cloud Based on Multi-Level Kriging Interpolation, Remote Sens. 8 (1) (2016) 35.
[28] K. Kraus, N. Pfeifer, Determination of terrain models in wooded areas with airborne laser scanner data, ISPRS J. Photogramm. Remote Sens. 53 (4) (1998) 193–203.
[29] P. Axelsson, DEM generation from laser scanner data using adaptive TIN models, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. XXXIII (Pt. B4/1) (2000) 110–117.
[30] H. Hu, Y. Ding, Q. Zhu, et al., An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy, ISPRS J. Photogramm. Remote Sens. 92 (2014) 98–111.
[31] G. Vosselman, Building reconstruction using planar faces in very high density height data, Int. Arch. Photogramm. Remote Sens. 32 (1999) 87–92.
[32] G. Vosselman, B.G.H. Gorte, G. Sithole, et al., Recognising structure in laser scanner point clouds, Int. Arch. Photogramm. Remote Sensing & Spatial Inf. Sci. 46 (Part 8/W2) (2004) 33–38.
[33] B. Yang, Z. Wei, Q. Li, et al., Semiautomated building facade footprint extraction from mobile LiDAR point clouds, IEEE Geosci. Remote Sens. Lett. 10 (4) (2013) 766–770.
[34] J. Zhang, X. Lin, Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification, ISPRS J. Photogramm. Remote Sens. 81 (2013) 44–59.
[35] G. Sithole, G. Vosselman, Filtering of airborne laser scanner data based on segmented point clouds, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. 36 (Part 3/W19) (2005) 66–71.
[36] X. Hu, X. Li, Y. Zhang, Fast filtering of LiDAR point cloud in urban areas based on scan line segmentation and GPU acceleration, IEEE Geosci. Remote Sens. Lett. 10 (2) (2013) 308–312.
[37] D. Tóvári, N. Pfeifer, Segmentation based robust interpolation–a new approach to laser data filtering, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci. 36 (Part 3/W19) (2005) 79–84.
[38] C. Chen, Y. Li, C. Yan, et al., An improved multi-resolution hierarchical classification method based on robust segmentation for filtering ALS point clouds, Int. J. Remote Sens. 37 (4) (2016) 950–968.
[39] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, New York, 2001.
[40] M. Bartels, H. Wei, D.C. Mason, DTM generation from LIDAR data using Skewness Balancing, 18th International Conference on Pattern Recognition I, 2006, pp. 566–569.
[41] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Software 3 (3) (1977) 209–226.
[42] M. Weinmann, S. Urban, S. Hinz, B. Jutzi, C. Mallet, Distinctive 2d and 3d features for automated large-scale scene analysis in urban areas, Comput. Graph. 49 (2015) 47–57.
[43] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. 39 (1) (1977) 1–38.
[44] A. Maguya, V. Junttila, T. Kauranne, Algorithm for extracting digital terrain models under forest canopy from airborne LiDAR data, Remote Sens. 6 (7) (2014) 6524–6548.