

A Fast Progressive TIN Densification Filtering Algorithm for Airborne LiDAR Data using Adjacent Surface Information

Hongfu Li, Chengming Ye, Zixuan Guo, Ruilong Wei, Lixuan Wang, Jonathan Li, *Senior Member, IEEE*

Abstract—Point cloud filtering is a preliminary and essential step in various applications of airborne LiDAR (light detection and ranging) data, with progressive triangulated irregular network (TIN) densification (PTD) being one of the classic methods for filtering LiDAR point clouds. The PTD algorithm densifies ground points through iteration operation based on initial ground seed points. However, the poor performance in steeply sloped areas and time-consuming processing are serious drawbacks for PTD algorithms. In this paper, we propose a fast progressive TIN densification (FPTD) filtering algorithm for airborne LiDAR data using adjacent surface information. After carefully establishing parameters and removing outliers, our improved FPTD uses a sliding window to obtain significantly more initial ground seed points. And we modified some iterative determination criterion, including the definition of maximum relative elevation threshold and the introduction of signed computation, to eliminate avoidable non-ground points. Then adjacent surface information was utilized to iterate each point cloud block, which is the smallest unit that point cloud can be segmented. Additionally, the algorithm is easily run in a multi-threaded environment, further accelerating the filtering process to some extent. Experiments show that our proposed FPTD filtering algorithm is fast and robust. Compared to the PTD, the FPTD algorithm yields better error rates and kappa coefficients in 1/12 of the time required by the PTD.

Index Terms—point cloud, sliding window, progressive TIN densification, adjacent surface information, fast filtering

I. INTRODUCTION

LiDAR (light detection and ranging) has become a popular remote sensing observation technique [1]. LiDAR systems fall into three main categories based on the data acquisition platform [2]: terrestrial laser scanning (TLS), mobile laser scanning (MLS), and airborne laser scanning (ALS). Of these, ALS use has grown significantly due to its ability to capture dense and accurate topographic data at high speed [3]. ALS has been widely employed in various fields [4], including digital terrain model (DTM) generation [5], [6],

This work was supported in part by the Second Tibetan Plateau Scientific Expedition and Research Program (STEP) under Grant 2019QZKK0902, the National Natural Science Foundation of China under Grant 42071411, and the key research and development program of Sichuan Province under Grant 22ZDYF2824. (Corresponding author: Chengming Ye)

H. Li, C. Ye, Z. Guo, R. Wei and L. Wang are with the Key Laboratory of Earth Exploration and Information Techniques of Ministry Education, Chengdu University of Technology, Chengdu, China (e-mail: leehungu@sina.com; rsgis@sina.com; gzxplus@sina.cn; viryloon@gmail.com; lixuanw2@gmail.com).

J. Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca)

forest ecosystem investigation [7]–[9], 3D building modeling [10]–[12], and geological disaster surveys [13]–[15]. In most LiDAR applications, separating point clouds into ground and object points is a preliminary but essential step for subsequent processing [16], with various filtering methods proposed for automatically extracting bare earth surface points. These filters fall into seven broad categories according to the underlying technique of mathematical morphology, iterative interpolation, TIN (triangular irregular network), CS (cloth simulation), slope, segmentation, or machine learning.

Several comparisons of filtering algorithms [17]–[19] have shown that TIN-based filters, primarily those using PTD (progressive triangulated irregular network densification) algorithms, offer strong and stable abilities when acquiring bare ground points. Many researchers have improved the PTD algorithm in three large ways. The first improvement concerns the initial ground seed points, for it is widely accepted that the more initial ground seed points, the better filtering results. To identify more initial ground seed points, Zhang and Lin [20] improved the PTD algorithm using a point cloud segmentation method called segmentation using smoothness constraint (SUSC). Specifically, after selecting the lowest points in each grid cell as initial ground seed points, SUSC expands the set of ground seed points as much as possible. Similarly, Xu and Yue [21] firstly segmented point cloud data into several clusters based on plane fitting, then, followed by a coarse spatial clustering process using dual distances to obtain a set of initial ground seed points. Zhao et al. [22] first acquired potential ground seed points using a morphological method, and then used a translation plane fitting method to identify and remove non-ground points from the potential points. By assuming measurements that collide with cloth particles are ground points, the cloth simulation filtering [23] acquired initial ground seed points [4] and subsequently generated a high-quality initial provisional digital terrain model (DTM). These research results show that plentiful and accurate initial ground seed points influence the filtering results to a large extent.

The second PTD improvement concerns the iterative determination criterion. Lin and Zhang [24] modified the PTD algorithm by processing a segment unit rather than a single point. If the number of ground measurements is larger than the number of object measurements in a single segment, all of the points within the current segment are labeled as ground measurements. This segment-based filtering method preserves discontinuities in landscapes and removes the lower

parts of large objects attached on the ground surface. To reduce the probability of incorrectly classifying non-ground points on the lower objects as ground points, only one point (the potential ground point with the shortest distance to the corresponding TIN facet) is selected as the ground point. The experimental results show that the revised PTD approach performs better than the PTD method in removing non-ground points, which are from low-to-the-ground objects [25]. And Shi et al. [26] proposed a parameter-free progressive TIN densification filtering algorithm based on the slope estimation.

The third improvement to the PTD is to shorten the processing time. Kang et al. [27] used state-of-the-art multi-core computing facilities to speed up the computationally intensive task by encapsulating the PTD filter into independent computing units with clearly faster results. However, this research ignores filtering accuracy. Other incremental improvements to the PTD were also made in [28], [29].

Most of the above studies point out that the performance of PTD algorithms is poor in steeply sloped areas. Because PTD algorithms rely on two thresholds (maximum angle and maximum distance [20]) and initial ground seed points to a large extent. Complex terrain will prevent PTD algorithms from extracting ground points through these fixed thresholds if there are not enough evenly-distributed initial ground seed points. Additionally, we find the PTD algorithm is very time-consuming [27] for locating the corresponding triangle facet of a iterative point is time consuming, especially while the amount of ground points increase, causing a large number of triangle networks. However, most triangles are far away the iterative point, and can not help extract ground points but only increase time consumption. Therefore, we propose a fast progressive TIN densification filtering algorithm (FPTD) for airborne LiDAR data using adjacent surface information. In this method, we first use the sliding window algorithm to select large numbers of evenly-distributed initial ground seed points. Next, the point cloud divider breaks the data into a series of minimal processing blocks equal in size to the sliding window. Then, we iterate unclassified points in each block based on adjacent surface information, which greatly accelerates the filtering process. What's more, the improved algorithm is easily run in a multi-threaded environment, which can accelerate the TIN densification process still further. This method notably avoids reductions in filtering accuracy and actually increases it with most datasets. There are three main contributions of our paper: (1) A large number of evenly-distributed initial ground seed points can be readily obtained using the sliding window algorithm. (2) A new filtering strategy based on adjacent surface information is proposed, which can greatly shorten the processing time and improve the filtering accuracy. (3) A method to optimize the threshold selection of PTD filtering algorithms by classifying datasets according to the terrain complexity is proposed.

We organize our paper as follows. We present our improved PTD algorithm in detail in Section II. Section III presents the results and discussion. Finally, Section IV gives our conclusions.

II. METHOD

Our proposed method contains five key steps: (1) specifying parameters, (2) removing outliers, (3) obtaining initial ground seed points, (4) segmenting the point cloud into blocks, and (5) iterating all unclassified points and densifying the TIN. The detailed procedure of the proposed FPTD algorithm is illustrated in Fig.1. The following subsections provide full details of each step.

A. Specifying parameters

Our proposed algorithm requires seven parameters to be preset:

(1) Radius threshold for outlier removal, r . This parameter eliminates outliers below the ground and retains sparse ground points in some areas. r should be appropriately big.

(2) Point number threshold for outlier removal, n . Similar to r , this parameter should be appropriately small.

(3) Maximum relative elevation threshold, e . Different from maximum distance [20], e is the relative elevation threshold from an unclassified point to the highest vertex in the corresponding triangle (seen in Fig. 2). If the elevation of an unclassified point minus that of the highest vertex is larger than e , the point will be labeled as the object point. Otherwise it will be labeled as a potential ground point.

(4) Maximum angle threshold, a . a is the angle threshold between the triangle plane and the line connecting an unclassified point to its corresponding closest triangle vertex [20] (seen in Fig. 2). If the angle is smaller than a , the point will be labeled as a potential ground point. Otherwise, it will be labeled as an object point.

(5) Sliding window size, w . w depends on the size of maximum artificial object (buildings, cars and so on). Specifically, to ensure the initial ground seed points are not obtained from building rooftops or other objects, w must be bigger than the maximum size of artificial objects in the point cloud dataset. . If w is smaller than the size of the biggest artificial objects in the point cloud, the lowest point in a sliding window may be from the objects, not the ground, causing false initial ground seed points. Therefore, to obtain more initial ground points, w should be as small as possible once it is greater than the biggest artificial object. Further, block size is always both equal to w for convenience in this paper.

(6) Sliding stride, s . s is a distance between two adjacent windows. It makes the window slide according to the stride distance.

(7) Neighborhood radius to the center point of a block point cloud, d . The value of d depends on the block size. It determines the initial ground seed points for a block point cloud, with $d = 1.5\sqrt{2}w$ providing a sufficient number in most cases. The sufficient initial ground seed points means that a block of data should contain the initial ground seed points along with those of its 8 neighborhood blocks.

B. Removing outliers

The PTD filtering algorithm performs well, relying on the hypothesis that there are no outliers with a lower height

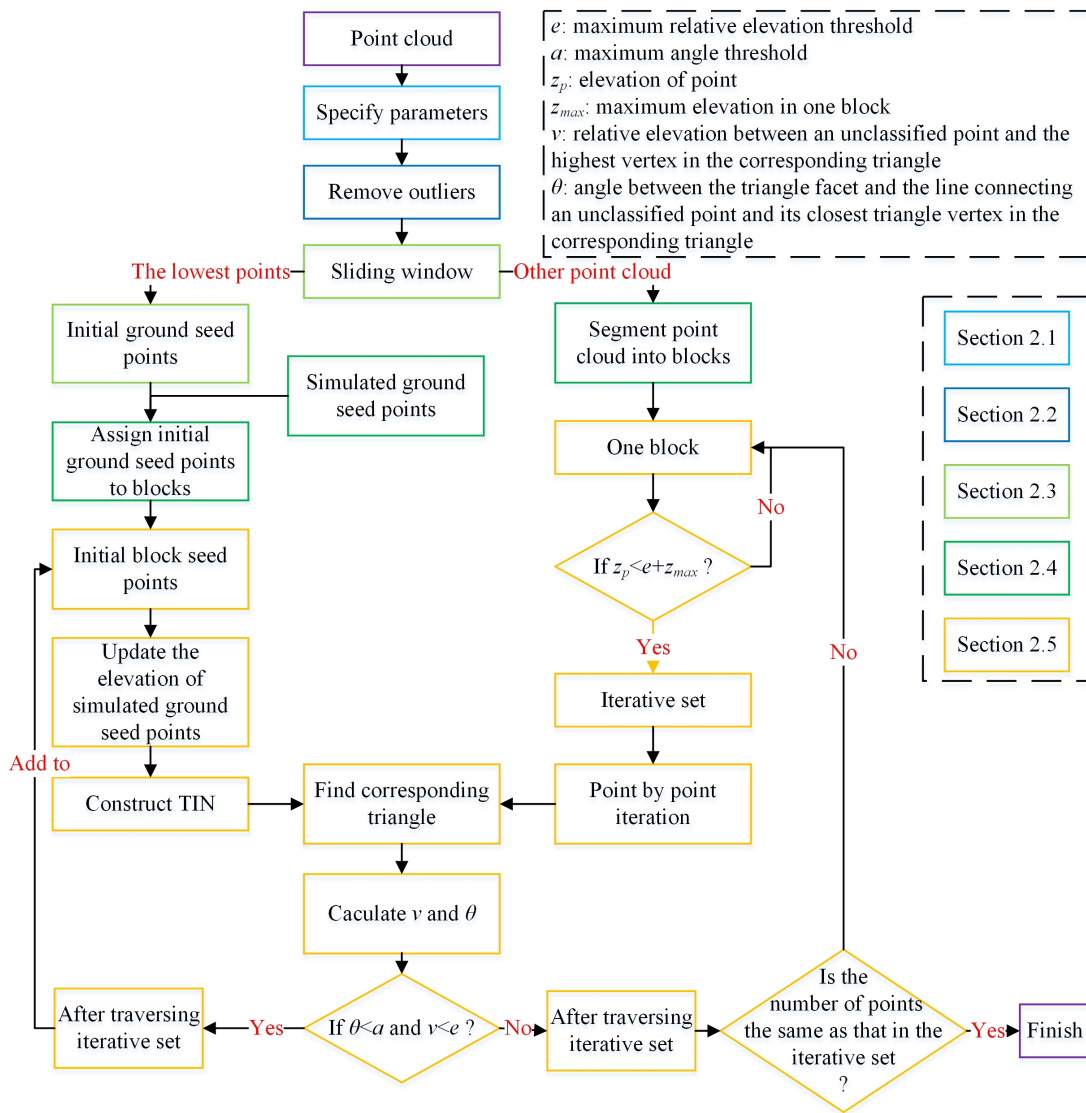


Fig. 1. Flow chart of our improved filtering method.

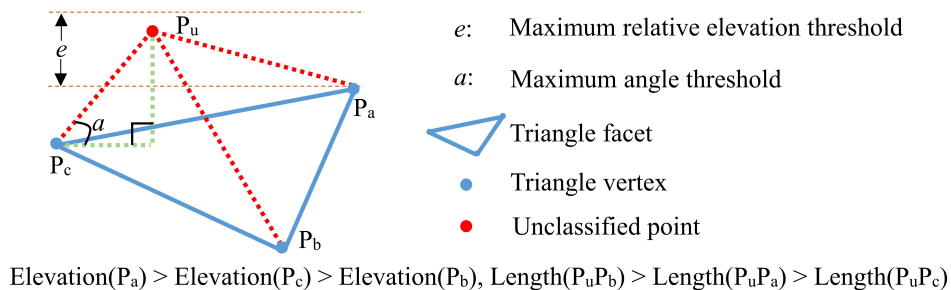


Fig. 2. The definition of the threshold parameter.

than the ground. If there are, the radius outlier removal algorithm can be used to eliminate these low outliers. The radius outlier removal algorithm iterates every point in point cloud, find the points (outliers) whose amount of points in a radius threshold (r) is less than the point number threshold (n), and remove these points out. After using the radius outlier removal algorithm, it is a good idea to check the data again

and remove the low outliers manually if any outliers remain.

C. Obtaining initial ground seed points

One of the most critical steps for PTD algorithms is obtaining accurate and appropriate initial ground seed points. We employ a new and simple strategy, the sliding window algorithm, to extract initial ground seed points (Fig. 3) with the

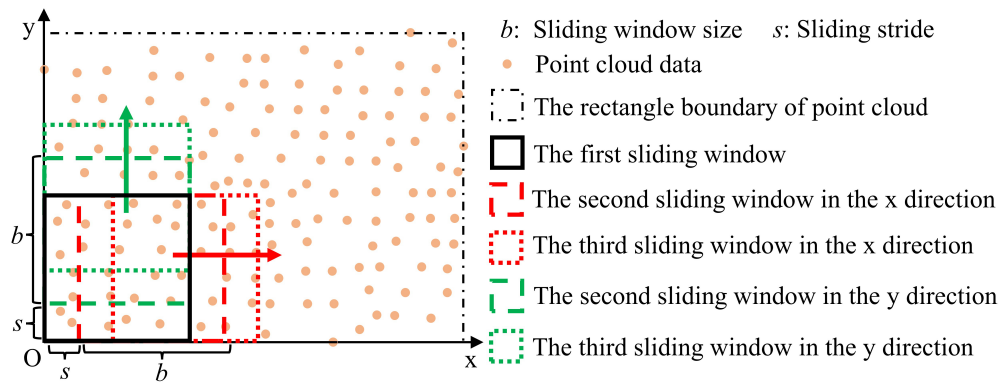


Fig. 3. Sliding window algorithm to obtain initial ground seed points.

following steps. First, choose a square window of size w and set the sliding stride s . Second, calculate the minimum values of x and y directions to obtain the rectangular boundary of the point cloud, with the origin at the lower left corner. Third, slide the window from left to right within the rectangular boundary using the slide stride increment while finding the lowest point in each window. The point with the lowest elevation in the window is regarded as initial ground seed point. Fourth, slide the window up along the y direction while repeating the third step. Fifth, repeat the previous step until the window has covered the entire point cloud. Compared to most improved PTD algorithms [4], [20]–[22], the sliding window algorithm is much easier to implement, and more importantly, the obtained initial ground seed points are more evenly-distributed.

Note that (1) the coordinates of datasets have been projected before extracting initial ground seed points. (2) If the size of the last evaluated section of the point cloud in the x or y direction is smaller than the slide stride, that last section will be added to the prior window, meaning that the last window size may be bigger than others.

D. Segmenting point cloud into blocks

The PTD algorithm extracts ground points well, but the time it requires to do so is a serious drawback. To reduce the processing time, we segment unclassified point clouds into a series of blocks with overlapping regions (shown in Fig. 4) and use each block as a minimal processing unit. The minimal processing unit means the block can not be segmented again, which can accelerate the filtering process to the greatest extent. Concretely, the block size is equal to that of sliding window w , with the width of overlap equal to a multiple of the sliding stride. In data segmentation, if the width of the last part of the point cloud in the x or y direction is smaller than block size, it will be added to the previous block.

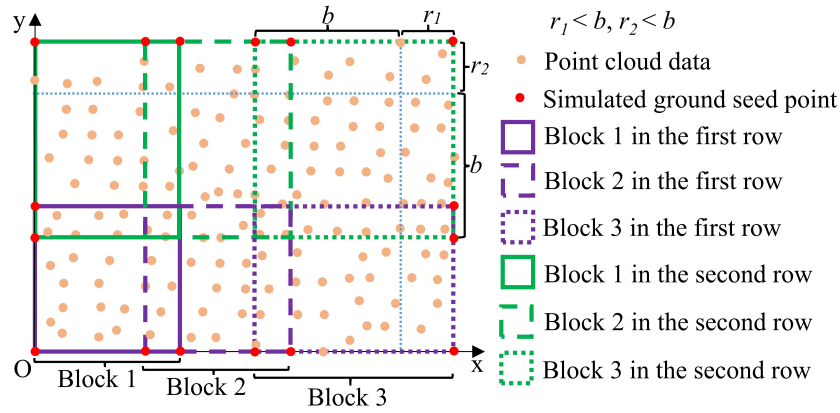
Next, the initial ground seed points (described in Section II-C) within a certain neighborhood distance (d) from the center of a block should be assigned to the corresponding block. These initial ground seed point are the initial block seed points in the next part. Notably, the value of d is extremely important, and its value must make sure the block to be processed is surrounded by adequate initial ground seed points.

So, d is equal to $d = 1.5\sqrt{2}w$, which means the initial ground seed points assigned to the block are from its corresponding 8 adjacent blocks. In this way, sufficient adjacent surface information can be utilized. To ensure that all of the points in each block will be located in the TIN constructed by the initial block seed points, simulated ground seed points (points highlighted in red in Fig. 4) are created for the block on the edge of the rectangle boundary before assigning initial ground seed points to blocks. The elevation of simulated ground seed points is the same as that of its nearest initial ground seed point. Afterwards, these simulated ground seed points are added to the initial ground seed point set, and then the new initial ground seed points are assigned to each block using the criteria above. Once a block and its corresponding initial block seed points are determined, we can employ multiple threads to process a large block point cloud.

E. Iterating unclassified points and densifying the TIN

In the PTD algorithm, all of the unclassified points are traversed based on the whole initial ground seed points in the densification process. However, each unclassified point only belongs to one corresponding triangle. Most triangles far from the iterative point have no effect on filtering results but consume a lot of processing time. And the more the ground points extracted, the more prominent this phenomenon becomes. Unlike the PTD, we only iterate the unclassified points in a block based on its corresponding initial block seed points. In this way, we preserve the adjacent surface information of points while reducing unnecessary processing.

The iteration process has five steps: (1) Within a single block, select the unclassified points having an elevation lower than the sum of the highest point in the initial block seed points and the maximum relative threshold. These are called the iterative unclassified points set, or the iterative set for short. (2) Choose an unclassified point in the iterative set and locate its corresponding triangle facet. (3) Calculate the plane equation (Equation (1)) according to the triangular vertices. (4) Calculate the two variables to be compared, which refer to the relative elevation from the unclassified point to the highest vertex of corresponding triangle, and the angle between the triangle facet and the line connecting the unclassified point and the closest triangle vertex, as seen in Equations (2) and (3).



Note: If the width of the last part of point cloud is smaller than block size (b), it will be added to the former block.

Fig. 4. Segmenting the point cloud data into blocks.

In this step, the two variables are slight different from those in the PTD. They are both signed. If their values are negative, it means the point is below the triangle facet and the point can also be identified as a ground point. In this way, it easily solves the problem mentioned in Section 2.3.1 of [25]. (4) Label the unclassified point. If the relative elevation and angle are both smaller than the corresponding threshold, the point will be labeled as a ground point. (5) Iterate all unclassified points in the iterative set and then update the elevation of the simulated ground seed points and the TIN according to the new ground points. Because the initial elevation of the simulated ground seed points are estimated from the corresponding nearest initial ground seed points, the estimated elevation may deviate far from the true adjacent ground surface with more and more ground points being extracted, resulting in that the ground points near the simulated ground seed points may not be extracted efficiently. Steps (1)–(5) are repeated until no unclassified points can be added to the ground point set.

By confining each iteration to a single block of the point cloud, multi-threaded techniques can be utilized to process multiple partitioned datasets simultaneously, shortening the processing time further to some extent.

$$A * x + B * y + C * z + D = 0 \quad (1)$$

$$v = z_u - z_m \quad (2)$$

$$\theta = \arcsin(d/l) \quad (3)$$

$$d = \frac{A * x_u + B * y_u + C * z_u + D}{\sqrt{A^2 + B^2 + C^2}}$$

$$l = \sqrt{(x_u - x_0)^2 + (y_u - y_0)^2 + (z_u - z_0)^2}$$

In these equations, A , B , C , and D are the parameters of the plane equation of a facet. v is the relative elevation between an unclassified point (x_u, y_u, z_u) and the highest vertex in the corresponding triangle. θ is the angle between a triangle facet

and the line connecting an unclassified point and its closest triangle vertex (x_0, y_0, z_0) in the corresponding triangle. z_m is the elevation of the highest vertex in the corresponding facet. d is the signed distance from an unclassified point to its corresponding facet, meaning that the point is under the facet if d is less than zero, and above the facet otherwise. l is the distance between an unclassified point and its closest triangle vertex.

III. RESULTS AND DISCUSSION

To analyze the performance of our proposed algorithm, we used benchmark datasets provided by ISPRS Working Group III/3, including 8 unlabeled datasets from different terrains (4 urban sites and 4 rural sites) with 15 labeled samples from these sites. The eight datasets are named: CSite1, CSite2, CSite3, CSite4, FSite5, FSite6, FSite7, and FSite8. The 15 labeled samples are: s11, s12, s21, s22, s23, s24, s31, s41, s42, s51, s52, s53, s54, s61, and s71. Because these datasets contain different terrains, it is difficult to determine the optimal thresholds of PTD algorithms. Therefore, as shown in Table I, we tentatively classified these data into four categories according to the terrain complexity (mainly the degree of slope change): flat (D1), flat with gentle slopes (D2), flat with scarp (D3), and steep (D4). For labeled samples, we calculated four indicators to evaluate the filtering results quantitatively as shown in Table II. We implemented our process on a desktop computer with an Intel Core i5-10400F CPU and 16 GB of RAM using Python under the Windows 10 operating system. We used eight threads for the iterative densification process.

A. Parameter analysis

All of the parameters (Section II-A) have different effects on the filtering results, but the maximum relative elevation threshold and maximum angle threshold matter the most. Thus, we paid particular attention to these two parameters. Referring to other experiments [20], [24], [25], we varied the maximum angle threshold range from 2 to 42 degree in 4 degree increments and the maximum relative elevation threshold from 0.6 to 2.0 m in 0.2 m increments. To evaluate

TABLE I
FOUR TERRAIN CATEGORIES AND THE CORRESPONDING THRESHOLDS ACCORDING TO THE TERRAIN COMPLEXITY

Abbreviation	Category	Data	a	e
D1	flat	s12/s21/s31/s42/s54/CSite2/CSite3/CSite4/FSite7	18	1.0
D2	flat with gentle slope	s11/s22/s51/s61/s71/CSite1/FSite6/FSite8	22	1.2
D3	flat with scarp	s23/s24/s41	30	0.8
D4	steep	s52/s53/FSite5	38	1.4

TABLE II
CALCULATION OF TYPE I ERROR (TI), TYPE II ERROR (TII), TOTAL ERROR (TE) AND KAPPA COEFFICIENT (K): a AND b ARE THE NUMBERS OF GROUND POINTS CLASSIFIED CORRECTLY AND INCORRECTLY, RESPECTIVELY; d AND c ARE THE NUMBERS OF NON-GROUND POINTS CLASSIFIED CORRECTLY AND INCORRECTLY, RESPECTIVELY

		Filtering		Metrics of Quantitative Evaluations		
		Ground	Object			
Reference	Ground	a	b	$f = (a + b)/e$	$TI = b/(a + b)$	$P_o = (a + d)/e$
	Object	c	d	$g = (c + d)/e$	$TII = c/(c + d)$	$P_e = f * h + g * i$
		$h = (a + c)/e$	$i = (b + d)/e$	$e = a + b + c + d$	$TE = (b + c)/e$	$K = (P_o - P_e)/(1 - P_e)$

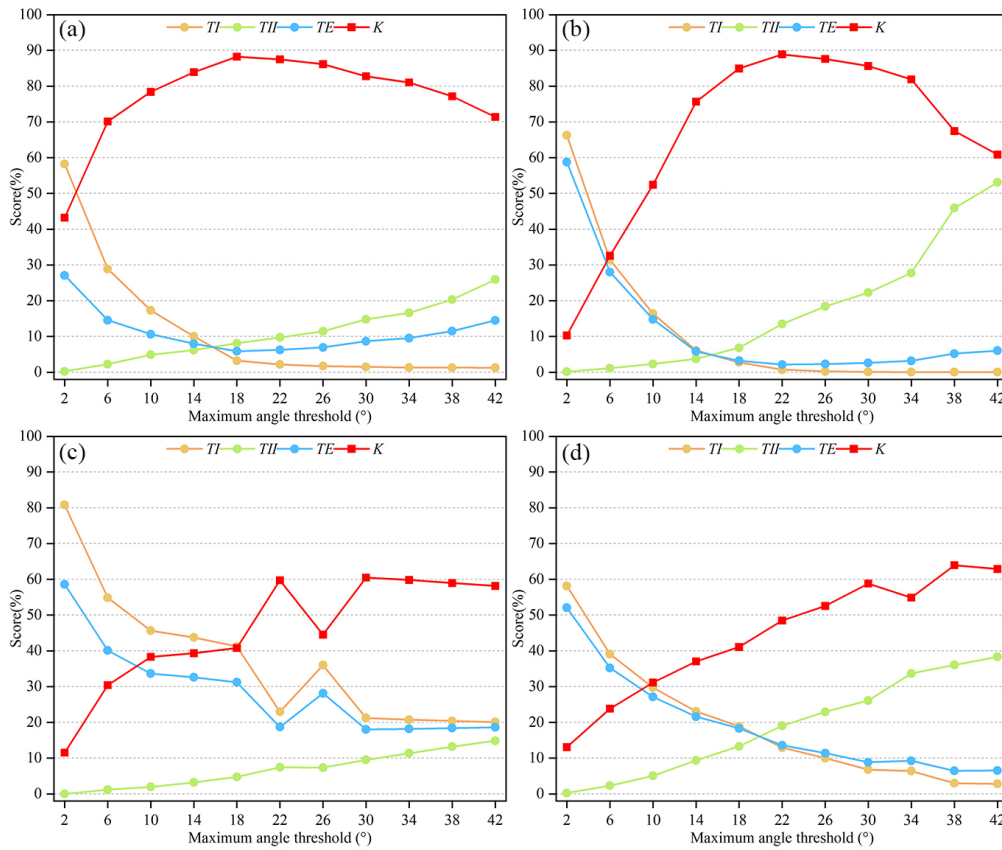


Fig. 5. Optimal maximum angle threshold analyses of different terrains: (a) for flat terrain (D1); (b) for flat with gentle slope (D2); (c) for flat with scarp (D3); and (d) for steep terrain (D4).

the filtering results using different parameters, we chose sub-regions with fewer points as the experimental data for each terrain category: s54 of D1, s71 of D2, s24 of D3, and s52 of D4. We set other parameters based on our experience: $r = 5$, $n = 10$, $w = 60$, and $s = 10$.

For the first stage, we analyzed the maximum angle threshold while holding the maximum relative elevation threshold at 1.4. As seen in Fig. 5, although the trends in Fig. 5(c) and (d)

are not particularly obvious, the kappa coefficient generally increased first and then decreased as the angle threshold increased, and the total error changed in the opposite direction. Additionally, the larger the maximum angle threshold, the smaller the type I error and the larger the type II error, indicating that more ground points were extracted correctly but more object points were misidentified at the same time. The optimal maximum angle thresholds were 18, 22, 30, and 38

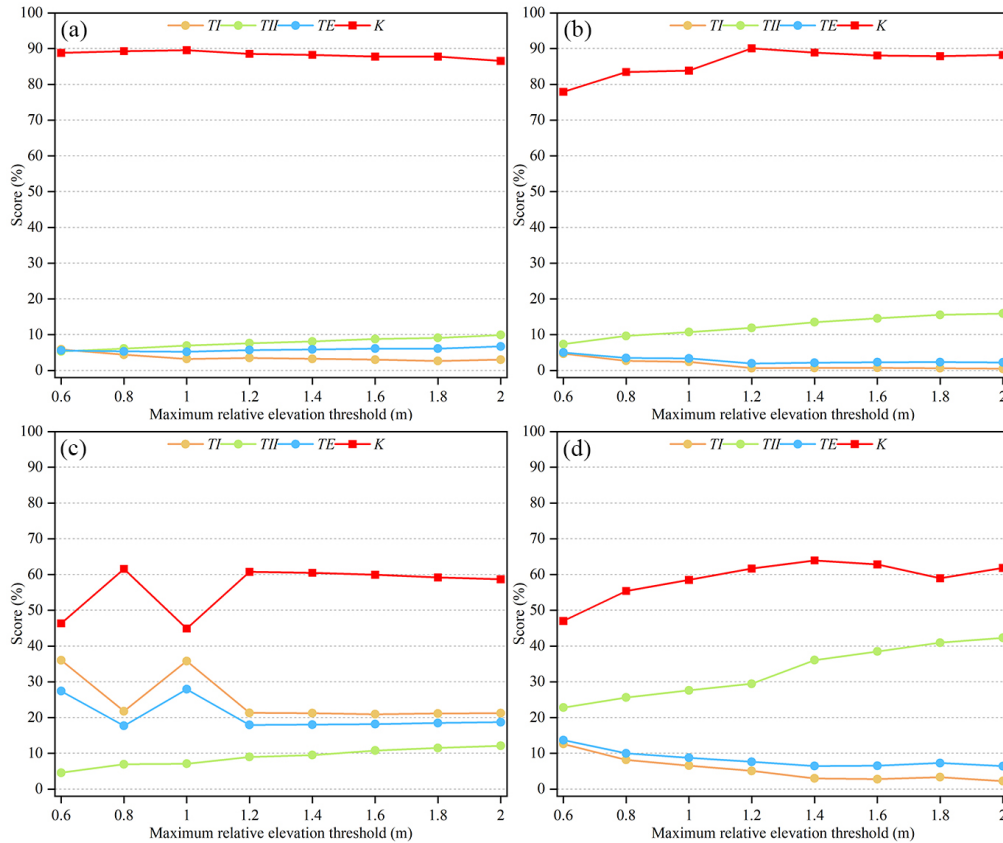


Fig. 6. Optimal maximum relative elevation threshold analyses of different terrains: (a) for flat terrain (D1); (b) for flat with gentle slope (D2); (c) for flat with scarp (D3); and (d) for steep terrain (D4).

for D1, D2, D3, and D4, respectively. From these thresholds, a conclusion can be drawn: the more complicated the terrain, the larger the optimal corresponding maximum angle threshold.

For the second stage, we examined the maximum relative elevation threshold in light of the optimal maximum angle threshold. The kappa coefficient in Fig. 6 fluctuated slightly less than that in Fig. 5, indicating that the FPTD filtering algorithm was more sensitive to the maximum angle threshold. Nonetheless, we were able to determine experimentally the optimal maximum relative elevation thresholds for different terrains: 1.0 for D1, 1.2 for D2, 0.8 for D3, and 1.4 for D4. Except for D3, the maximum relative elevation threshold increased with the terrain complexity. Both the optimal maximum angle and relative elevation thresholds of different terrains are listed in Table I. Our subsequent experiments were based on these optimal thresholds.

B. Results

After specifying the optimal thresholds, we applied these values to all of the unlabeled datasets and obtained the filtering results shown in Table III. Although the number of extracted ground points did not show much useful information, the time cost was significantly reduced by the improved filtering algorithm. We have provided a number-time diagram (Fig. 7) to show the relationship between the total number of points and the processing time, using a linear function to fit these two

variables. The processing time varied linearly with the number of points, indicating the high and robust processing efficiency.

High efficiency is usually accompanied by poor extraction results. Thus, to evaluate the performance of our algorithm, we selected the filtering results for FSite5, a relatively complex terrain, for visualization. Fig. 8 shows the overall high performance in our experiments. The buildings in the left area were eliminated robustly, and the vegetation in the lower right corner was also noticeably filtered out. In Fig. 8, (a-1) and (b-1) are vertical cross-sections of (a) and (b) in red highlight. In the same figure, (a-2) and (b-2) are horizontal cross-sections of (a) and (b), respectively, showing more specifically that the filtering results are consistent with the actual terrain.

C. Performance evaluation between the classic and improved PTD methods

We have worked to improve the PTD algorithm in three aspects: (1) the method of selecting initial ground seed points, (2) the iterative judgement criterion, and (3) the processing time. The filtering results in Section III-B indicate the improved PTD algorithm was effective and efficient. However, the differences in the results from the PTD and our improved PTD methods were obscure, so we tested the filtering performance of these two methods on labeled samples. The PTD was implemented according to [20]. Four different criteria were calculated, as given in Table IV. The data in the table shows that the improved PTD filter earned a lower *TI* score, but a

TABLE III
FILTERING RESULTS OF THE EIGHT UNLABELED DATASETS

Dataset	CSite1	CSite2	CSite3	CSite4	FSite5	FSite6	FSite7	FSite8
Total number of points (points)	1366408	486800	377028	518060	628576	551698	393264	345966
Number of ground points (points)	689324	254277	193514	262193	525191	460188	347944	294724
Time cost (s)	104.80	27.68	17.52	23.54	31.48	26.10	21.17	18.11

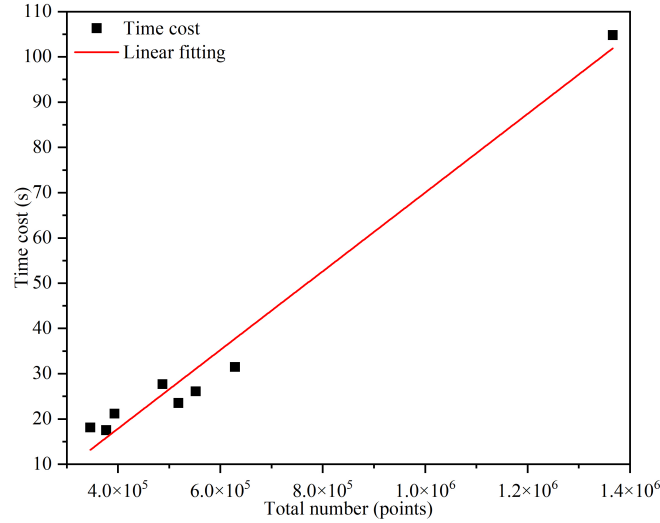


Fig. 7. The relationship between the total number of points and the corresponding time cost.

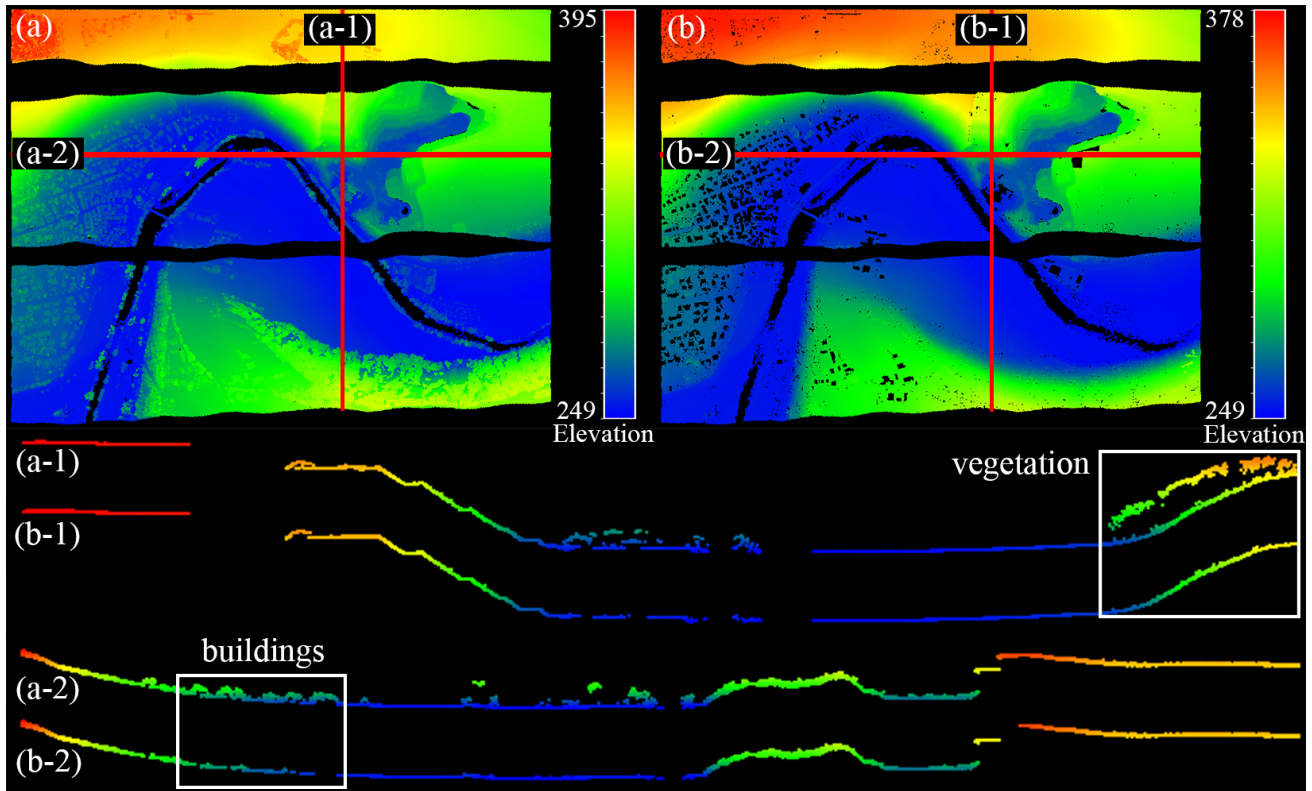


Fig. 8. The filtering results for FSite5 using the optimal thresholds of D4: (a) original data rendered by elevation with (a-1) and (a-2) as the corresponding cross-sections; (b) filtering result using the FPTD rendered by elevation with (b-1) and (b-2) as the corresponding cross-sections. The black area is the background.

higher *TII* score than the PTD, indicating that our method extracted the ground points more reliably but generated more

TABLE IV
QUANTITATIVE EVALUATION OF THE FILTERING RESULTS USING THE TWO ALGORITHMS

Terrain	Dataset	Method	Total number ¹	<i>TI</i> (%)	<i>TII</i> (%)	<i>TE</i> (%)	<i>K</i> (%)	Time cost (s)
D1	s12	FPTD	52119	9.86	4.01	7.01	86	3.70
		PTD	52119	10.25	2.06	6.26	87.5	55
	s21	FPTD	12960	0.91	3.97	1.59	95.38	1.81
		PTD	12960	2.6	3.9	2.89	91.79	3.46
	s31	FPTD	28862	0.7	3.28	1.89	96.19	2.13
		PTD	28862	1.5	2.73	2.06	95.84	16.50
	s42	FPTD	42470	1.45	1.44	1.44	96.54	2.15
		PTD	42470	4.62	1.37	2.32	94.37	64.07
	s54*	FPTD	8608	3.21	6.94	5.22	89.54	1.62
		PTD	8608	16.75	3.35	9.55	80.62	6.66
	s11	FPTD	38010	10.87	15.46	12.83	73.75	7.68
		PTD	38010	25.46	8.74	18.32	63.73	75.58
s22	FPTD	32706	10.89	5.77	9.29	79.38	3.14	
	PTD	32706	9.83	6.9	8.92	80.03	28.38	
D2	s51	FPTD	17845	0.06	24.31	5.35	82.83	2.01
		PTD	17845	7.29	13.53	8.65	75.75	32.6
s61	FPTD	35060	2.84	8.79	3.05	65.84	3.23	
	PTD	35060	4.26	3.9	4.25	58.94	44.24	
s71*	FPTD	15645	0.66	11.92	1.94	90.06	1.96	
	PTD	15645	1.12	7.51	1.85	90.85	9.23	
s23	FPTD	25095	12.74	5.3	9.22	81.59	2.48	
	PTD	25095	23.78	3.65	14.25	71.74	75.06	
D3	s24*	FPTD	7492	21.79	6.95	17.71	61.59	2.31
		PTD	7492	41.3	5.15	31.37	40.49	7.98
s41	FPTD	11231	3.84	2.4	3.12	93.77	1.69	
	PTD	11231	58.69	0.64	29.6	40.72	4.82	
s52*	FPTD	22474	2.98	36.07	6.46	63.96	2.98	
	PTD	22474	9.62	26.29	11.37	51.46	25.22	
D4	s53	FPTD	34378	3.99	46.87	5.72	39.99	3.41
		PTD	34378	16.35	27.14	16.79	20.7	52.11
All	Average	FPTD	25664	5.79	12.23	6.12	79.76	2.82
		PTD	25664	15.56	7.79	11.23	69.63	33.39

¹ Total number: Total number of points in dataset (points)

* These datasets were used to analyse the optimal parameters

object points at the same time. Therefore, it is difficult to determine which filtering algorithm was better in terms of *TI* and *TII*. From the overall indicators *TE* and *K*, our improved algorithm performed better in most cases. More specifically, there were 12 datasets where the filtering results using the proposed algorithm were better than those using the PTD, while three datasets (s12, s22, and s71) showed slightly worse performance from our improved algorithm. The average total error of our FPTD was 6.12, while that of the PTD is nearly 2 times higher. The average kappa coefficient of the FPTD was almost 10% higher than that of the PTD. It is also worth mentioning the much shorter processing time required by our algorithm. For all of the labeled samples, the processing time is significantly reduced. In addition, the average processing time is nearly shortened by 12 times, and even nearly 30 times in some datasets (s42, s23).

Although the performance of our proposed algorithm was generally robust and efficient, there were some areas with poor filtering results. To explore the limitations in our method,

we selected s12, s61, s24, and s53 for detailed analysis. The filtering results for these four regions were poorer than others, and each comes from a different type of terrain: s12 from D1, s61 from D2, s24 from D3, and s53 from D4. As shown in Figs. 9(b), 10(b), 11(b), and 12(b), the initial ground seed points obtained by two methods were markedly different. There were more initial ground seed points using our FPTD algorithm than the PTD. Most of the initial ground seed points acquired by the PTD were included as points acquired by the FPTD, apart from several simulated ground seed points. Thus, the FPTD readily obtained more dense initial ground seed points.

For s12 (Fig. 9), both the PTD and the FPTD performed terribly, especially in the white highlighted area. We classified this dataset into D1 because most areas of it are flat. However, there is a noticeable change in the topography (highlighted in white) that caused the optimal threshold to be unsuitable for the area. We tested this area using the optimal thresholds from D3, a more complex terrain, and obtained much better

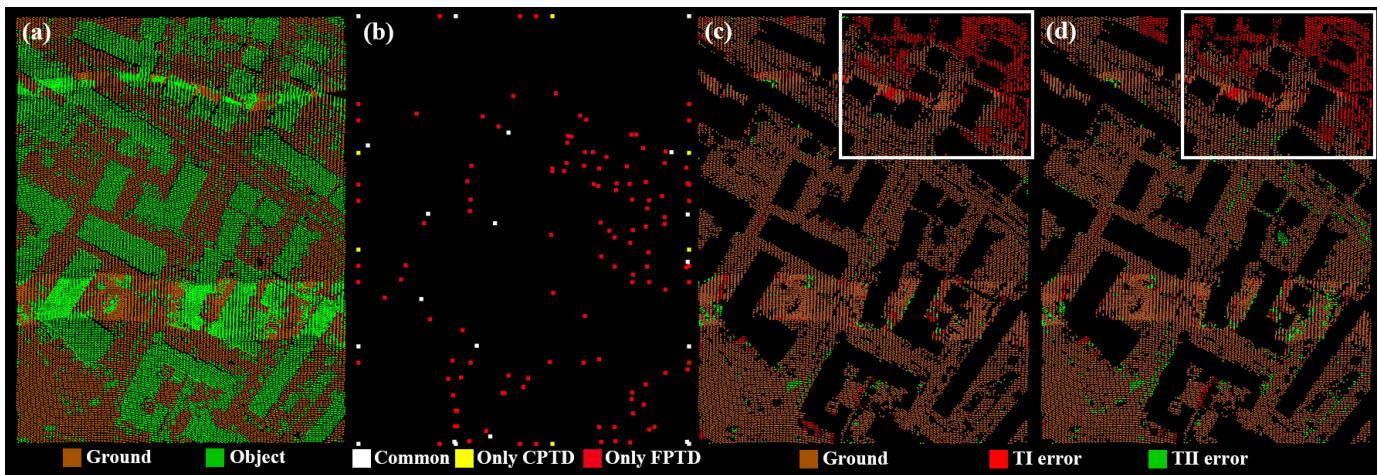


Fig. 9. The filtering results for s12 using two methods: (a) the reference data; (b) the initial ground seed points obtained by two methods (the PTD obtained 26; the FPTD obtained 140); (c) the results using the PTD; and (d) the results using the FPTD.

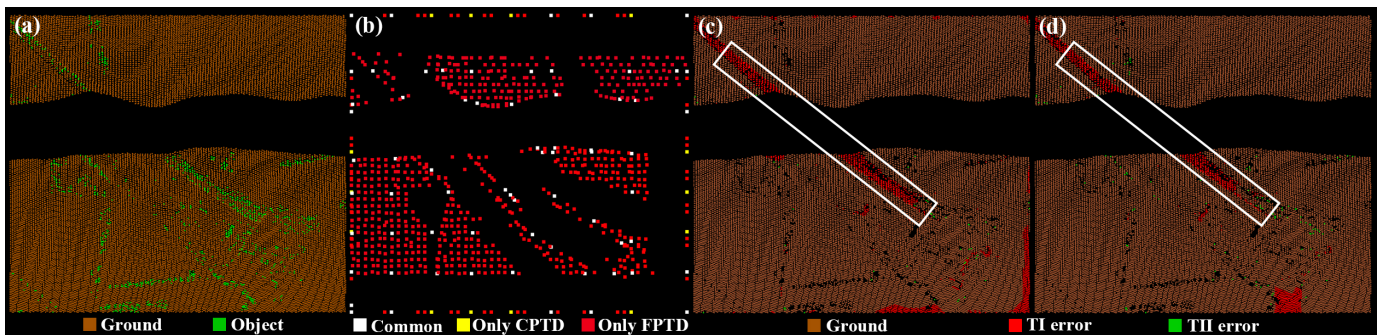


Fig. 10. The filtering results for s61 using two methods: (a) the reference data; (b) the initial ground seed points obtained by two methods (the PTD obtained 81; the FPTD obtained 820); (c) the results using the PTD; and (d) the results using the FPTD.

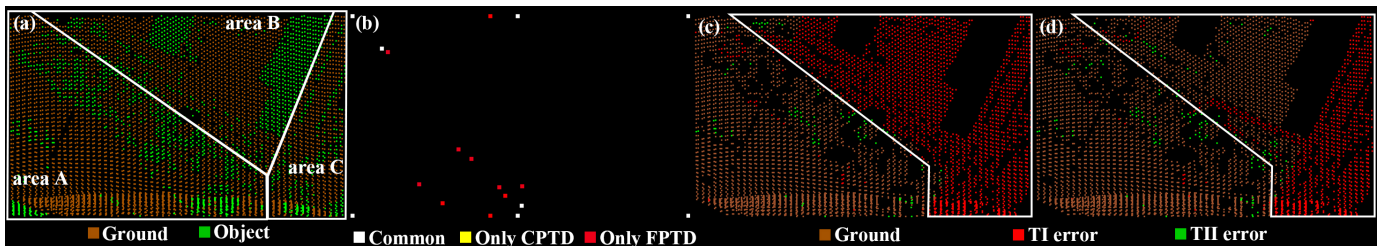


Fig. 11. The filtering results for s24 using two methods: (a) the reference data, with elevation (area A) ; elevation (area B) ; elevation (area C); (b) the initial ground seed points obtained by the two methods (the PTD obtained 8; the FPTD obtained 18); (c) the results using the PTD; and (d) the results using the FPTD.

filtering results. The scores of TI , TII , TE , and K using our proposed algorithm were 1.58%, 5.67%, 3.58%, and 92.83%, respectively. K was 6.83% higher than the original one. Notably, it is not that the larger the thresholds, the better the results. The scores of TI , TII , TE and K using the optimal thresholds from D4 were 0.61%, 12.44%, 6.38%, and 87.2%, indicating a drop in accuracy. For region s61 (Fig. 10), the same phenomenon occurred and the scores of TI , TII , TE , and K using the optimal thresholds of D4 were 0.96%, 29.02%, 1.93%, and 70.73%, respectively, indicating a better result. From these two datasets, we conclude that the datasets for mixed terrains should be classified based on the most complex area.

As shown in Fig. 11, most points in the right side (highlighted in white) of the dataset were filtered out using both the PTD and the FPTD, causing a high value of TI . There are two reasons for this. The first is that the dataset was too small, with a 121 m by 72 m rectangular boundary, resulting in incomplete continuity of the terrain and very few initial ground seed points. The block-wise operation magnified the problem. The second is that this dataset has noticeably abrupt changes in terrain. According to elevation, this dataset could be divided into three parts (Fig. 11(a)): the lower left corner (area A), the top part (area B), and the right part (area C). The elevation of area C is the highest, but the initial ground seed points are concentrated in area A, making it hard to identify the ground

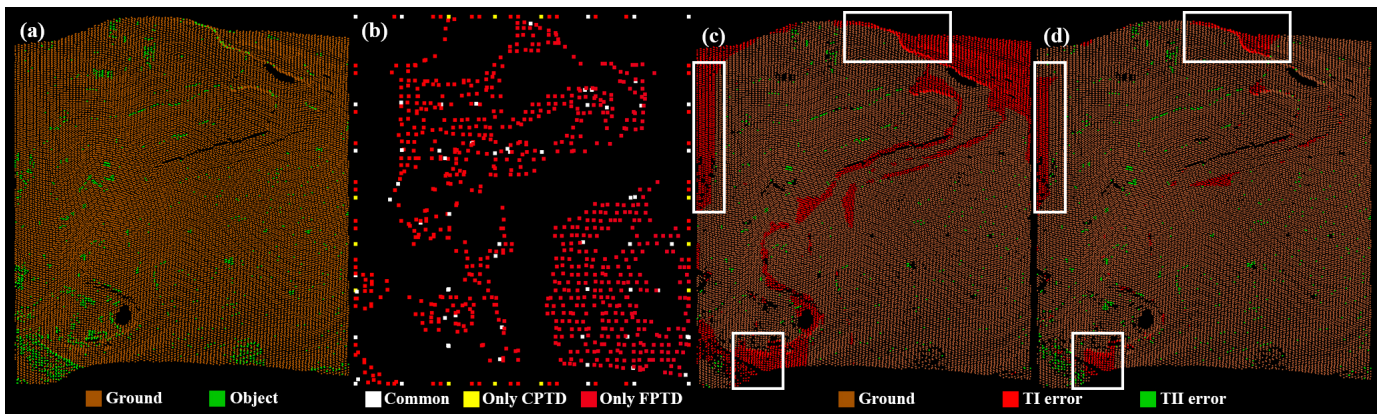


Fig. 12. The filtering results for s53 using two methods: (a) the reference data; (b) the initial ground seed points obtained by two methods (the PTD obtained 77; the FPTD obtained 841); (c) the results using the PTD; and (d) the results using the FPTD.

points in areas B and C. However, the FPTD produced results that were still superior to the PTD's due to the larger number of initial ground seed points.

An example of datasets with steep terrain is shown in Fig. 12. To some extent, both the PTD and the FPTD algorithms failed to extract ground points in steep areas, and this drawback is common to most PTD-based algorithms. However, our proposed filtering performed much better than the PTD due to the larger number of initial ground seed points (Fig. 12 (b)).

In addition, because the sum of the maximum terrain angle [20] plus maximum angle [20] was greater than 90 degrees, some object points with high elevation were identified as ground points using the PTD. The use of the maximum relative elevation threshold can help solve this problem.

IV. CONCLUSION

Due to the poor performance in steeply sloped areas and the time-consuming processing of classic PTD, we improved the PTD algorithm in three perspectives: the method of selecting initial ground seed points, the iterative judgement criterion, and the processing time. Experiments show our proposed FPTD filtering algorithm is fast and robust. Its average processing time was 1/12 of the PTD, and its average kappa coefficient of the FPTD was almost 10% higher than that of the PTD. Additionally, we draw three additional conclusions:

(1) Using the sliding window algorithm, we can readily obtain a large number of evenly-distributed initial ground seed points. Decreasing the window size and stride size produces more initial ground seed points.

(2) Segmenting datasets into blocks and assigning the initial ground seed points in a certain neighborhood to blocks guarantees the surface spatial relationship between blocks and greatly shortens the processing time.

(3) The classification of datasets according to the terrain complexity is very helpful for determining the thresholds of PTD filtering algorithms.

REFERENCES

- [1] C. Ye, J. Li, H. Jiang, H. Zhao, L. Ma, and M. Chapman, "Semi-Automated Generation of Road Transition Lines Using Mobile Laser Scanning Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1877–1890, may 2020.
- [2] K. C. Roberts, J. B. Lindsay, and A. A. Berg, "An analysis of ground-point classifiers for terrestrial LiDAR," *Remote Sensing*, vol. 11, no. 16, 2019.
- [3] S. Dayal, S. Goel, B. Lohani, N. Mittal, and R. K. Mishra, "Comprehensive Airborne Laser Scanning (ALS) Simulation," *Journal of the Indian Society of Remote Sensing*, 2021.
- [4] S. Cai, W. Zhang, X. Liang, P. Wan, J. Qi, S. Yu, G. Yan, and J. Shao, "Filtering airborne LiDAR data through complementary cloth simulation and progressive TIN densification filters," *Remote Sensing*, vol. 11, no. 9, may 2019.
- [5] Z. Chen, B. Devereux, B. Gao, and G. Amable, "Upward-fusion urban DTM generating method using airborne Lidar data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 72, pp. 121–130, aug 2012.
- [6] A. H. Ozcan and C. Unsalan, "LiDAR data filtering and DTM generation using empirical mode decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 360–371, jan 2017.
- [7] M. N. Bazezew, Y. A. Hussin, E. H. Kloosterman, I. M. Mohd Hasmadi, T. Soromessa, and M. S. Adan, "Factual approach for tropical forest parameters measurement and monitoring: future option with a focus on synergetic use of airborne and terrestrial LiDAR technologies," *International Journal of Remote Sensing*, vol. 42, no. 9, pp. 3219–3230, 2021.
- [8] Y. Shi, T. Wang, A. K. Skidmore, S. Holzwarth, U. Heiden, and M. Heurich, "Mapping individual silver fir trees using hyperspectral and LiDAR data in a Central European mixed forest," *International Journal of Applied Earth Observation and Geoinformation*, vol. 98, p. 102311, jun 2021.
- [9] W. Xu, S. Deng, D. Liang, and X. Cheng, "A crown morphology-based approach to individual tree detection in subtropical mixed broadleaf urban forests using UAV lidar data," *Remote Sensing*, vol. 13, no. 7, apr 2021.
- [10] G. A. Nys, F. Poux, and R. Billen, "City json building generation from airborne LiDAR 3D point clouds," *ISPRS International Journal of Geo-Information*, vol. 9, no. 9, sep 2020.
- [11] J. Song, S. Xia, J. Wang, and D. Chen, "Curved Buildings Reconstruction from Airborne LiDAR Data by Matching and Deforming Geometric Primitives," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1660–1674, feb 2021.
- [12] N. Yastikli and Z. Cetin, "Classification of raw LiDAR point cloud using point-based methods with spatial features for 3D building reconstruction," *Arabian Journal of Geosciences*, vol. 14, no. 3, feb 2021.
- [13] M. Syzdykbayev, B. Karimi, and H. A. Karimi, "Persistent homology on LiDAR data to detect landslides," *Remote Sensing of Environment*, vol. 246, sep 2020.
- [14] A. Tiwari, A. B. Narayan, R. Dwivedi, O. Dikshit, and B. Nagarajan, "Monitoring of landslide activity at the Sirobagarh landslide, Uttarakhand, India, using LiDAR, SAR interferometry and geodetic surveys," *Geocarto International*, vol. 35, no. 5, pp. 535–558, apr 2020.
- [15] J. Miandad, M. M. Darrow, M. D. Hendricks, and R. P. Daanen, "Landslide mapping using multiscale LiDAR digital elevation models," *Environmental and Engineering Geoscience*, vol. 26, no. 4, pp. 405–425, nov 2020.

- [16] X. Meng, N. Currit, and K. Zhao, "Ground filtering algorithms for airborne LiDAR data: A review of critical issues," pp. 833–860, mar 2010.
- [17] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 1-2, pp. 85–101, aug 2004.
- [18] B. Suleymanoglu and M. Soycon, "Comparison of filtering algorithms used for dtm production from airborne lidar data: A case study in Bergama, Turkey," *Geodetski Vestnik*, vol. 63, no. 3, pp. 395–414, 2019.
- [19] V. Moudrý, P. Klápště, M. Fogl, K. Gdulová, V. Barták, and R. Urban, "Assessment of LiDAR ground filtering algorithms for determining ground surface of non-natural terrain overgrown with forest and steppe vegetation," *Measurement: Journal of the International Measurement Confederation*, vol. 150, jan 2020.
- [20] J. Zhang and X. Lin, "Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 81, pp. 44–59, 2013.
- [21] Y. Xu and D. J. Yue, "A spatial clustering filtering method for airborne lidar point cloud based on dual distance," *Lasers in Engineering*, vol. 34, no. 1-3, pp. 167–181, 2016. [Online]. Available: https://www.researchgate.net/publication/308675830_A_spatial_clustering_filtering_method_for_airborne_lidar_point_cloud_based_on_dual_distance
- [22] X. Zhao, Q. Guo, Y. Su, and B. Xue, "Improved progressive TIN densification filtering algorithm for airborne LiDAR data in forested areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 79–91, jul 2016.
- [23] W. Zhang, J. Qi, P. Wan, H. Wang, D. Xie, X. Wang, and G. Yan, "An easy-to-use airborne LiDAR data filtering method based on cloth simulation," *Remote Sensing*, vol. 8, no. 6, 2016.
- [24] X. Lin and J. Zhang, "Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments," *Remote Sensing*, vol. 6, no. 2, pp. 1294–1326, feb 2014.
- [25] S. Nie, C. Wang, P. Dong, X. Xi, S. Luo, and H. Qin, "A revised progressive TIN densification for filtering airborne LiDAR data," *Measurement: Journal of the International Measurement Confederation*, vol. 104, pp. 70–77, jul 2017.
- [26] X. Shi, H. Ma, Y. Chen, L. Zhang, and W. Zhou, "A parameter-free progressive TIN densification filtering algorithm for lidar point clouds," *International Journal of Remote Sensing*, vol. 39, no. 20, pp. 6969–6982, oct 2018.
- [27] X. Kang, J. Liu, and X. Lin, "Streaming progressive TIN densification filter for airborne LiDAR point clouds using multi-core architectures," *Remote Sensing*, vol. 6, no. 8, pp. 7212–7232, 2014.
- [28] Y. Dong, X. Cui, L. Zhang, and H. Ai, "An improved progressive TIN densification filtering method considering the density and standard variance of point clouds," *ISPRS International Journal of Geo-Information*, vol. 7, no. 10, oct 2018.
- [29] H. Ma, W. Zhou, and L. Zhang, "DEM refinement by low vegetation removal based on the combination of full waveform data and progressive TIN densification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 260–271, dec 2018.



Chengming Ye received the Ph.D. degree in Earth exploration and information technology from the Chengdu University of Technology, Chengdu, China, in 2011. He is currently an Associate Professor with the College of Geophysics, Chengdu University of Technology. He was a Visiting Scholar with the Department of Geography and Environmental Management, University of Waterloo, Canada, from 2016 to 2017. He has published more than 20 articles in peer-reviewed journals including the *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS)*. His main research interests include geo-hazard remote sensing applications, ecological remote sensing, and LiDAR data processing.



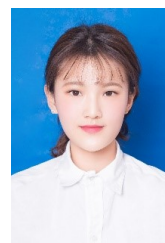
Zixuan Guo received the B.Eng. degree in space science and technology from the Chengdu University of Technology, China, in 2021, where she is currently pursuing the master's degree in Earth exploration and information technology. Her main research interests comprise geo-hazard remote sensing applications, and point cloud processing.



Ruilong Wei received the B.Eng. degree in Spatial information and digital technology from the Chengdu University of Technology, Chengdu, China, where he is currently working toward the MA.Eng. degree in Earth exploration and information technology. His research interests include Geo-hazard risk assessment, machine learning, and remote sensing applications.



Hongfu Li received the B.Eng. degree in space science and technology from the Chengdu University of Technology, China, in 2019, where he is currently pursuing the master's degree in Earth exploration and information technology. His main research interests comprise geo-hazard remote sensing applications, point cloud processing, hyperspectral image processing and deep learning.



Lixuan Wang received the B.Sc. degree in geoscience from Hubei University of Arts and Science, China, 2019. She is currently pursuing the master's degree in Earth exploration and information technology in Chengdu University of Technology. Her main research interests comprise Ecological and geohazard remote sensing application.



Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa. He is currently a Professor and the Head of the Mobile Sensing and Geodata Science Group, Department of Geography and Environmental Management, University of Waterloo, Canada. He has co-authored more than 400 publications, more than 200 of which were published in refereed journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), the IEEE TRANS-

ACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), ISPRS-JPRS, and RSE. His research interests include information extraction from LiDAR point clouds and from earth observation images. He is the Chair of the ISPRS WG I/2 on LiDAR, Air- and Space-borne Optical Sensing from 2016 to 2020 and the ICA Commission on Sensor-Driven Mapping for the period of 2019–2023, and the Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and Canadian Journal of Remote Sensing.