

Article

An Improved Point Cloud Filtering Algorithm Applies in Complex Urban Environments

Guangyu Liang ¹, Ximin Cui ^{1,*} , Debao Yuan ^{1,2}, Liuya Zhang ¹ and Renxu Yang ¹

¹ College of Geoscience and Surveying Engineering, China University of Mining and Technology-Beijing, Beijing 100083, China; sqt2300204082@student.cumtb.edu.cn (G.L.); yuandb@cumtb.edu.cn (D.Y.); sqt2200204094@student.cumtb.edu.cn (L.Z.); sqt2200204091@student.cumtb.edu.cn (R.Y.)

² Inner Mongolia Research Institute, China University of Mining and Technology-Beijing, Ordos 010300, China

* Correspondence: cxm@cumtb.edu.cn

Abstract: Point cloud filtering plays a crucial role in ground point extraction in urban environments. It can effectively distinguish ground points from object points, reduce data redundancy, and improve processing efficiency, providing accurate foundational data for urban 3D modeling, environmental monitoring, and intelligent management. However, current point cloud filtering algorithms have significant limitations in multi-scale structural complexity and sparse-to-dense balancing, hindering accurate extraction in complex urban environments. To address those issues, this paper proposes a point cloud filtering algorithm based on cloth simulation and progressive TIN densification (CAP). The algorithm first applies the cloth simulation filtering (CSF) algorithm to perform an initial filtering of the point cloud data and extract the initial ground points. It then constructs a TIN model based on the initial ground points, incorporating the concept of the progressive TIN densification (PTD) algorithm. Through point-by-point thresholding, the ground and object points are further refined and optimized. In the urban public point cloud datasets provided by ISPRS, the average total error is 5.90% after CAP algorithm filtering. For 12 sets of point cloud data in the North Rhine–Westphalia experimental sample area, the results show that the CAP algorithm achieves an average total error of 2.86%, which is 2.01% lower than the PTD algorithm and 0.60% lower than the CSF algorithm. The average Kappa coefficient is 94.04%, which is an improvement of 4.17% and 1.22% over the PTD and CSF algorithms, respectively. This study demonstrates that the CAP algorithm exhibits superior accuracy and adaptability for point cloud filtering tasks in urban environments, with significant application potential.

Keywords: point cloud filtering; ground point extraction; urban 3D modeling; CAP



Academic Editors: Shengbiao Wu and Pablo Rodríguez-González

Received: 23 February 2025

Revised: 1 April 2025

Accepted: 16 April 2025

Published: 18 April 2025

Citation: Liang, G.; Cui, X.; Yuan, D.; Zhang, L.; Yang, R. An Improved Point Cloud Filtering Algorithm Applies in Complex Urban Environments. *Remote Sens.* **2025**, *17*, 1452. <https://doi.org/10.3390/rs17081452>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the acceleration of urbanization, especially in areas where high-density buildings, complex road networks, and artificial facilities are interwoven, the extraction of ground and object points has become a core task in point cloud data processing [1–3]. In these complex environments, traditional ground point extraction methods are often limited by building occlusion, terrain changes, and interference from object points, resulting in insufficient accuracy and robustness [4–6]. Especially in densely built areas or irregular terrain areas, it is difficult to achieve ideal results with traditional algorithms, such as the progressive triangular irregular network encryption algorithm (PTD) [7,8]. Therefore, how to improve the accuracy of point cloud filtering, especially in complex urban environments, has become a hot topic of current research.

Existing point cloud filtering methods can be roughly divided into several categories [9]: TIN-based [10–12], surface-based [13,14], morphological-based [15–17], segmentation- and cluster-based [18–21], statistical analysis [22], and multi-scale comparison [23]. The PTD algorithm, a TIN-based method with gradual refinement, is widely used in point cloud filtering due to its ability to enhance accuracy and surface detail processing. It improves ground point extraction by constructing an irregular triangular mesh and progressively encrypting mesh density [24]. In recent years, scholars have improved the filtering accuracy and efficiency by integrating multiple methods. Li et al. [25] combined irregular triangulation with region growing; Sui et al. [26] applied regional segmentation and mathematical morphology; Li et al. [27] used point cloud block iteration acceleration; Shi et al. [28] proposed a non-parametric TIN densification algorithm.

In order to solve the problem of efficiency and ease of use of point cloud data processing in complex terrain, the point cloud filtering algorithm based on cloth simulation (CSF) has received extensive attention in recent years. The algorithm realizes data filtering by simulating the movement of cloth in the physical environment. The process starts by inverting a LiDAR point cloud and placing a rigid cloth over it. The positions of the cloth nodes are then used to approximate the ground surface. Finally, the ground points are identified by comparing the original LiDAR data with the generated surface [29]. Different from the traditional progressive triangulation encryption algorithm, the CSF method does not need a complex mesh construction and refinement process but automatically identifies through physical simulation, which significantly improves the computational efficiency and simplifies the operation process. However, CSF has limitations in complex terrains and environments. Li et al. [30] improved CSF for complex geomorphology; Cai et al. [31] proposed ICSF for better terrain reference; Yang et al. [32] developed BCSF for automatic filtering of multibeam bathymetry; Xue et al. [33] enhanced classification accuracy using ICSF and weighted weakly correlated random forest.

In urban environments, both the PTD and the CSF algorithms exhibit significant limitations. The PTD algorithm faces challenges primarily due to the sparse distribution of initial seed points, which results in inadequate accuracy in complex terrains or densely built urban settings [34]. This sparsity hinders the acquisition of sufficiently detailed seed point distributions, particularly in areas with uneven ground point distribution or high ground point density, thereby limiting the algorithm's precision improvement capabilities [35]. Additionally, the iterative refinement of the triangular mesh in PTD leads to high computational and storage overheads, especially when processing large-scale point cloud datasets in high-density urban environments, significantly affecting the algorithm's real-time performance and efficiency. On the other hand, while the CSF algorithm offers advantages in terms of ease of use and computational efficiency, it struggles to maintain accuracy and real-time performance in large-scale urban environments. The CSF algorithm relies on the assumption of smoothness in point cloud data, which is often violated in urban settings due to irregularities caused by building corners, road bumps, and tree structures. Furthermore, CSF faces difficulties in handling objects and details across different scales. For instance, in highly complex urban scenarios featuring intricate building facades and narrow streets, the algorithm's ability to process details at varying scales is compromised, leading to reduced simulation accuracy [36].

To solve the above problems, a method combining CSF pre-filtering and static TIN modeling is proposed in this study, which aims to improve the accuracy and efficiency of point cloud processing in complex urban environments through algorithm collaborative optimization and process reconstruction. The CAP leverages the simplicity and efficiency of CSF while incorporating the progressive refinement concept of PTD. The CAP algorithm first employs CSF to filter the point cloud, extracting more accurate ground points. Subse-

quently, a TIN model is constructed based on the filtered ground points. Compared to the classical PTD algorithm, the CAP approach reduces the number of encryption iterations, thereby improving computational efficiency. Additionally, an angular and distance-based judgment mechanism is introduced to further distinguish between ground points and object points. This mechanism optimizes the CSF algorithm by minimizing the aggregation of ground points within the object point set, enhancing the detail processing of object points, and ultimately improving the overall accuracy of ground point extraction.

2. Materials and Methods

2.1. Data

The data (shown in Figure 1) used in this study were sourced from North Rhine-Westphalia (NRW), Germany, and consisted of high-precision three-dimensional measurement data obtained through Airborne Laser Scanning (ALS) technology [37]. The dataset was captured by a laser scanner installed on an aircraft, which is capable of accurately capturing spatial information of the ground surface and the objects above it, including buildings, vegetation, and other complex terrain features.

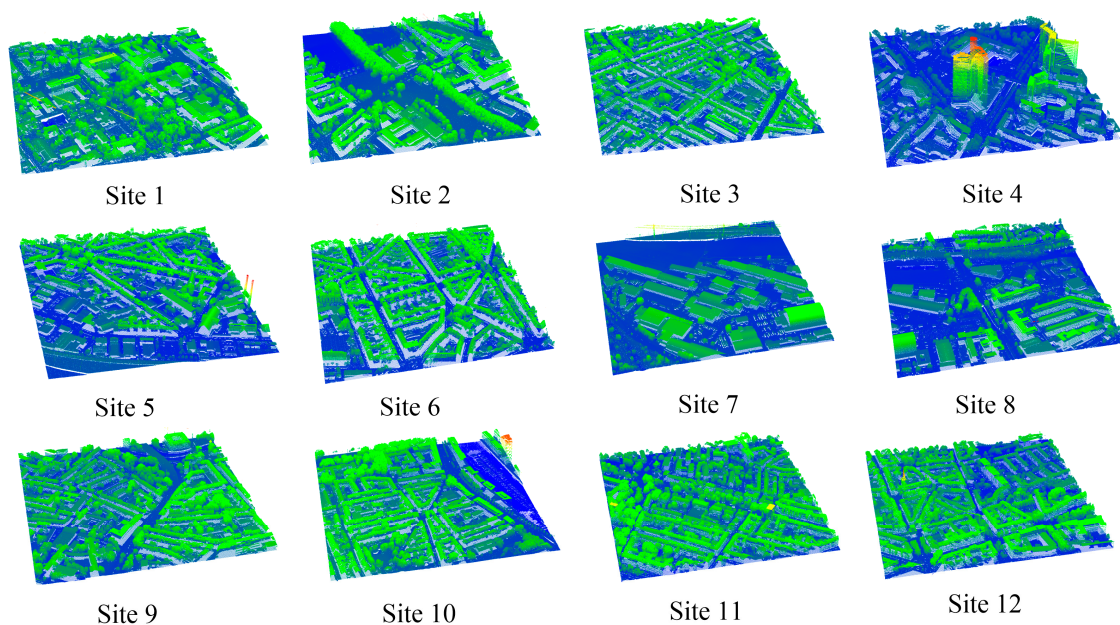


Figure 1. Original datasets of 12 experimental areas in North Rhine-Westphalia (NRW), Germany.

In order to verify the applicability and comparison accuracy of the algorithm, this study also used the International Society for Photogrammetry and Remote Sensing (ISPRS) public dataset for experiments and selected 9 samples of urban areas in the dataset for quantitative evaluation (Site 1–4).

2.2. Methodology

As shown in Figure 2, in this study, the combined algorithm mainly included three core steps: initial ground point acquisition, TIN model construction, and detailed selection of ground points. Firstly, based on the mass-spring model, the preprocessed point cloud was grided by using the CSF, the force state of each node was calculated step by step, and the node position was optimized by iteration to obtain the ground point. Then, these ground points were used as the basis for subsequent TIN model construction, and the initial TIN was generated by the Delaunay triangulation method. Finally, the remaining points were judged point by point, and whether the three angles formed by the vertical distance

between the checkpoint and the corresponding triangle plane and the triangle vertex line met the threshold conditions of distance and angle was filtered, so as to accurately distinguish ground points and target points, optimize the extraction accuracy of ground points, significantly reduce the probability of misclassification, and enhance the detail processing of ground points. The visualization diagram of the method process can be seen in Figure 3.

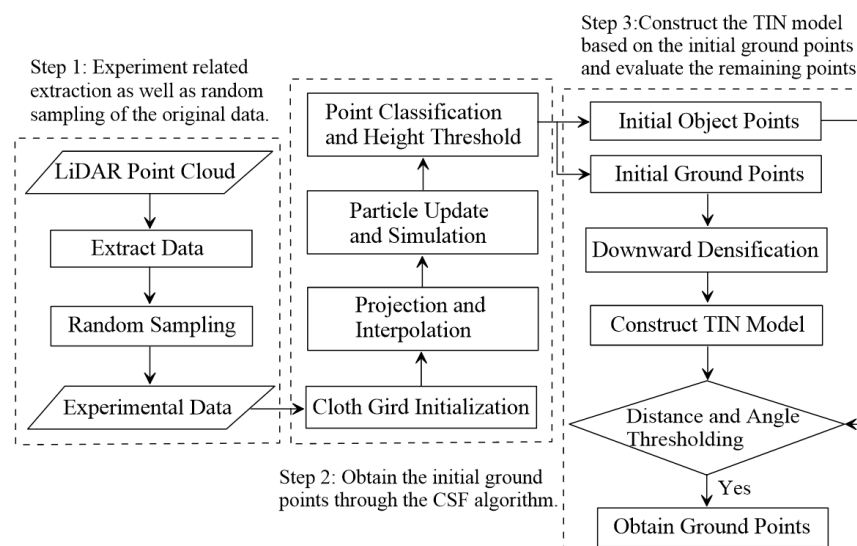


Figure 2. Flowchart of the research method.

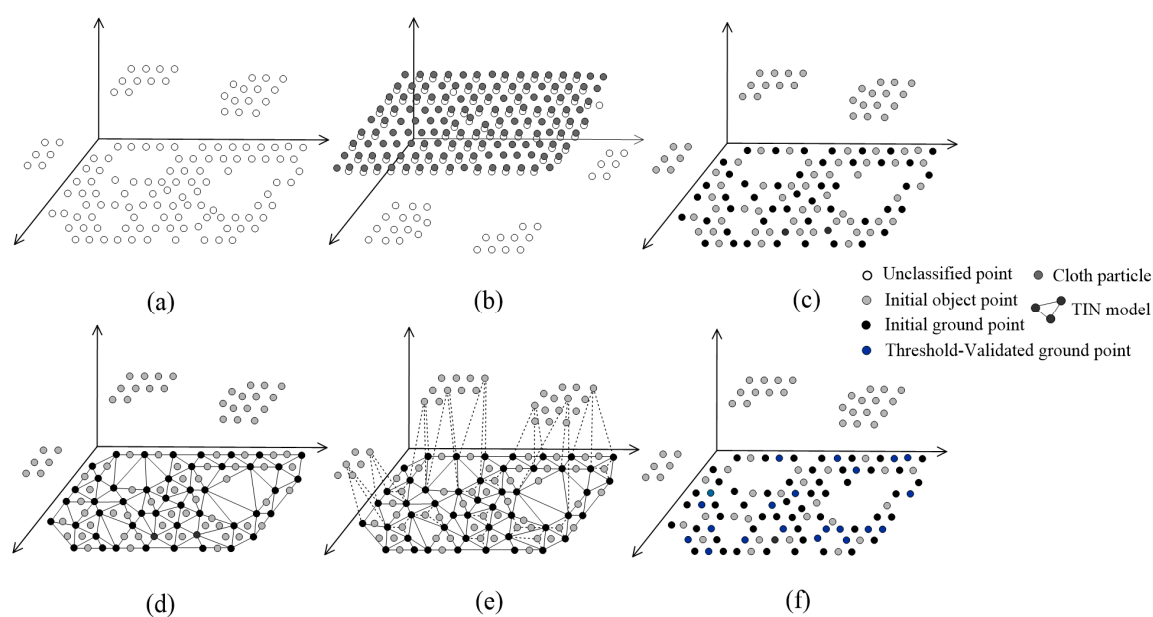


Figure 3. Schematic diagram of the algorithm flow. (a) Original point cloud data. (b) The CSF algorithm inverts the original data, and the particles move under the influence of internal and external forces. (c) Iterative steps are taken to obtain the initial ground points and initial object points. (d) The initial ground points obtained by the CSF algorithm are used to construct the TIN model. (e) The remaining points are traversed to judge the angle and distance thresholds. (f) The threshold-validated ground points are updated to obtain the final result.

2.2.1. Data Extraction and Preprocessing

The extraction of ground points and object points formed the basis of the analysis. Ground points primarily reflect the natural terrain surface, including ground elevation

points that are not affected by buildings, vegetation, or other man-made structures. Water bodies, such as lakes and rivers, are also classified as ground points to represent the natural undulations of the terrain. Object points, on the other hand, typically represent the reflective points of objects above the ground, such as buildings, trees, vehicles, and other structures, and correspond to the last return signals of the laser. The experimental data of last return ground points and last return (above the ground, e.g., buildings, trees) were selected as the experimental data, and the truth labels of ground points and object points were preset, so as to provide a basis for the full verification of the experimental results.

In order to improve the computational efficiency and ensure the representability of the results, random sampling of point cloud data was carried out for 12 urban sample areas with an area of roughly $500 \times 500 \text{ m}^2$, as shown in Figure 4, which was obtained by plane projection from the Z value of point cloud coordinates, and the specific elevation values are shown in the bar chart on the left. The random sampling ratio was set at 5% [38], a proportion that effectively preserves the spatial distribution characteristics and classification accuracy of the original data. This strategy significantly reduced the computational load while maintaining high classification accuracy and projection quality, thus meeting the analytical needs of this study.

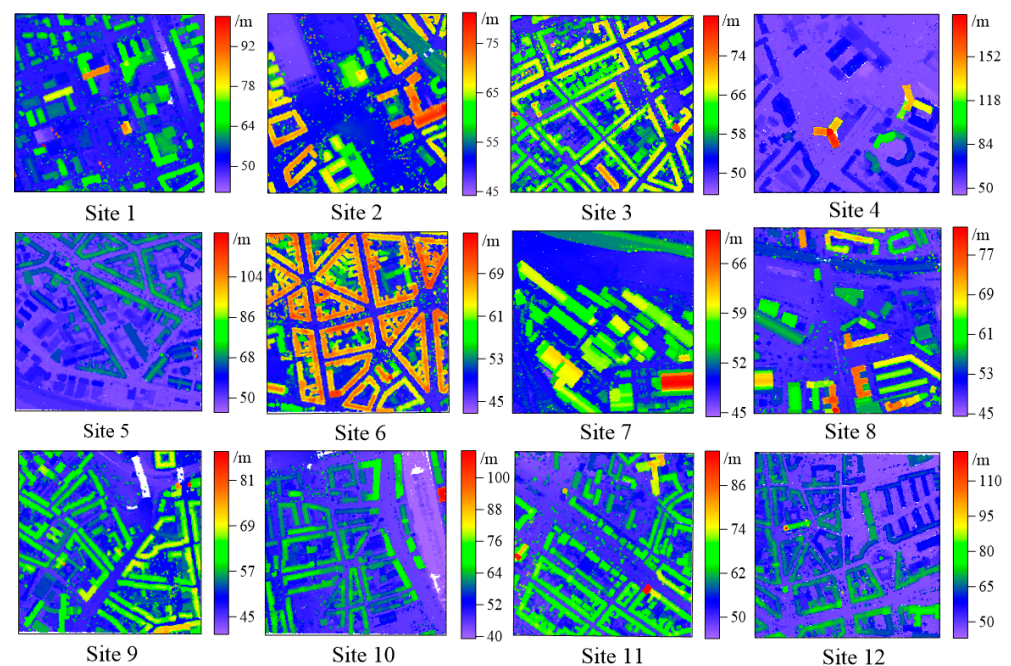


Figure 4. The experimental area was preprocessed and projected in a plane according to altitude.

2.2.2. Obtain Initial Ground Points

Since many scholars have carried out special theoretical and application research in this aspect [31,39,40], this paper only briefly describes the processing steps and the corresponding parameter explanation of this process of point cloud. Figure 5 shows the filtering process of the CSF algorithm.

The cloth simulation process starts by inverting preprocessed point cloud data and initializing the cloth mesh. The grid resolution (GR) determines the number of particles and their layout, while the cloth's rigidity (RI) is classified into three types to suit different terrains. The cloth is placed above the highest point of the point cloud to avoid ground contact. During projection, each particle is mapped onto a horizontal plane, and the closest LiDAR point (corresponding point or CP) is identified, with its height recorded as the intersection height value (IHV). Gravity and internal forces affect the particle positions, and this process repeats until the maximum iteration limit (MI) is reached. After simulation,

cloud-to-cloud distances are used to classify stable ground points and object points with significant changes, using a threshold (h) for differentiation. For steep slopes, post-processing smooths edges for accurate classification, adjusting particles based on height differences with neighboring particles. A steep fit factor (ST) ensures proper post-processing for steep slopes and unbiased traversal through strongly connected components (SCCs).

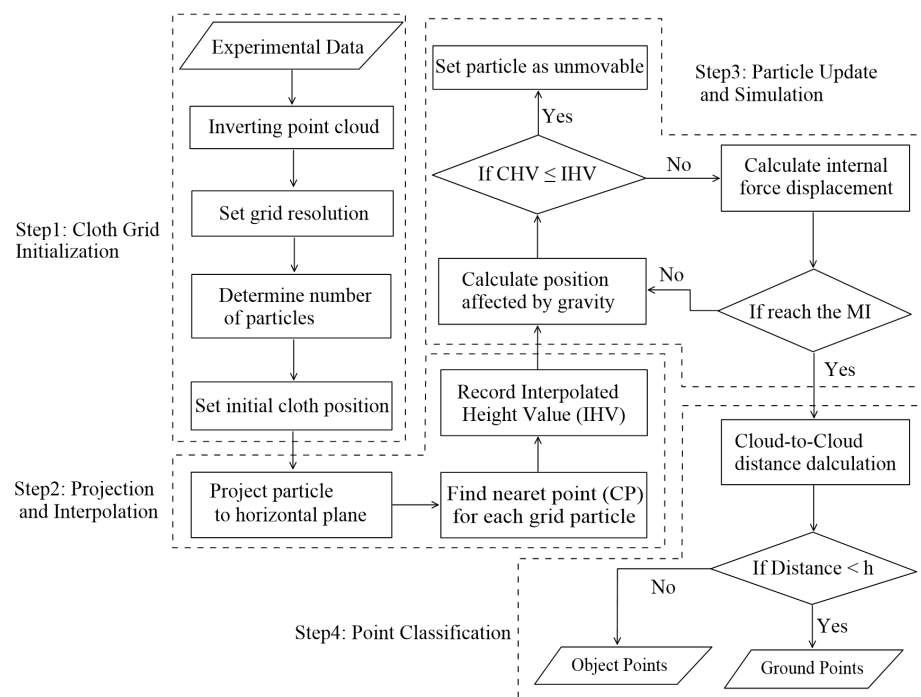


Figure 5. CSF algorithm flowchart.

2.2.3. Construct the TIN and Perform Threshold Judgement

The PTD algorithm is a point cloud filtering method based on TIN, widely used for extracting ground points from LiDAR data. This algorithm improves the accuracy of ground point extraction by progressively densifying the initial TIN model, adding new ground points and optimizing the mesh structure in each iteration [41].

In this study, inspired by the PTD algorithm, an irregular TIN model was constructed, and ground points were progressively selected and densified.

As shown in Figure 6b, the ground seed points extracted by CSF algorithm were used to complete the construction of TIN model. Compared with the random construction of initial points in the traditional PTD algorithm (a), this process greatly improves the accuracy and density of initial points and effectively reduces the densification process of TIN model iteration. Based on these ground points, with the help of the C++ platform, CGAL, PCL and other open source libraries, the initial TIN was constructed by using the Delaunay triangulation method. In this process, the ground points obtained by CSF filtering were used as the basis for grid construction. An initial set of triangular facets was formed by connecting adjacent ground points. This lays the foundation for the subsequent classification of remaining points and ensures the effectiveness and accuracy of the CAP algorithm in the process of ground point extraction.

In the point cloud filtering process, the processing of unclassified points usually involves analyzing the geometric relationship between the point cloud and the ground model, as shown in Figure 7. After constructing the TIN model, the remaining points were processed one by one. The classification process was as follows: with the help of the Fade2.5D open source library, the remaining points were located to their TIN grid, and then the remaining points were filtered according to the vertical distance from the point to

the TIN triangle. Then, the points were further classified by calculating the angle between the line between the unclassified points and the triangle vertices and the plane formed by the triangle. To ensure accuracy, this paper set a strict threshold condition, that is, on the basis of satisfying the distance threshold condition, the three included angles should also meet the set angle threshold. This two-step filtering method helps to refine the ground point extraction and improve accuracy and precision.

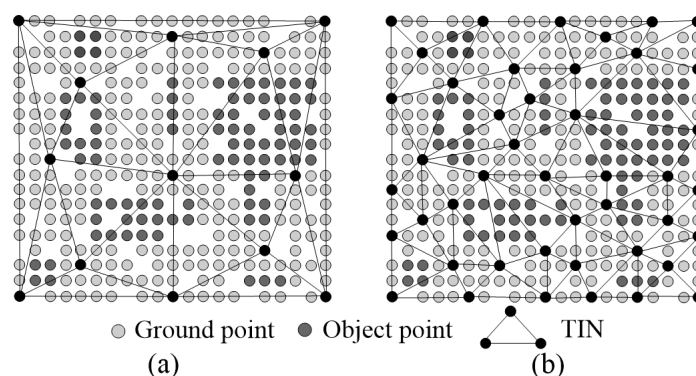


Figure 6. Schematic of the TIN model. (a) TIN model constructed from initial seed points of the original PTD algorithm; (b) TIN model constructed from ground points obtained using the CSF algorithm.

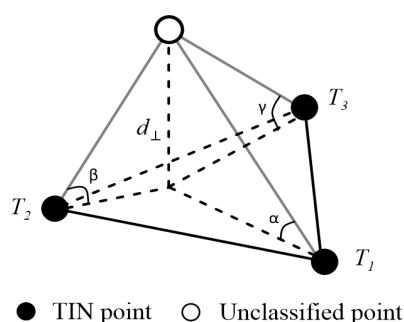


Figure 7. Distance and angle from unclassified points to the TIN model.

2.3. Parameter Setting

Parameters are crucial for the performance of an algorithm, and different parameter settings often lead to significant differences in results. In this study, various parameter configurations were applied based on different experimental regions. For the ISPRS official public dataset, the parameters for the CSF algorithm were set following the work of Zhang et al. [29], with an angle threshold of 6° and a distance threshold of 0.8 m. For the NRW experimental area, the following parameter settings were used: RI = 1, GR = 1.5 m, ST = True, h = 0.3 m, with the angle and distance thresholds remaining the same. A detailed sensitivity analysis of the parameters is provided in Section 3.4. The classification standards and accuracy metrics methods are shown in Tables 1 and 2.

Table 1. Classification standards for experimental data [42].

Reference	Ground Object	Filtered	
		Ground <i>a</i> <i>c</i>	Object <i>b</i> <i>d</i>

Table 2. Calculation methods for accuracy metrics [42].

Metrics of Quantitative Evaluations	
e	$a + b + c + d$
TI	$b / (a + b)$
TII	$c / (c + d)$
TE	$(b + c) / e$
P_o	$(a + b) / e$
P_c	$((a + b) \times (a + c) + (c + d) \times (b + d)) / e^2$
$Kappa$	$(P_o - P_c) / (1 - P_c)$

3. Results

3.1. Comparative Analysis with Other Algorithms Using the ISPRS Dataset

To better illustrate the effectiveness of the filtering algorithm in this paper, the filtering results of the ISPRS public dataset were compared with those of other algorithms. We selected 7 emerging filtering algorithms [11,12,14,28,43–45]. The total errors for urban scenarios are shown in Table 3. In sample 21, the algorithm achieves the lowest total error of 1.87%, while in sample 11, the total error is relatively high at 18.04%, which significantly lowers the average accuracy. Compared to the other seven algorithms, the algorithm proposed in this paper performs better in terms of accuracy, with an average total error of 5.90%, ranking third among all eight algorithms. Compared to recent filtering algorithms (such as Cai, Wang, etc.), the proposed algorithm shows significant improvement in filtering accuracy. This indicates that, although there is still room for improvement in certain specific scenarios (such as steep slopes and areas with significant terrain variations), its overall performance has made notable progress compared to existing technologies. In general, the algorithm not only has good precision performance but also has high adaptability and reliability, which is suitable for most urban environments.

Table 3. Total errors of emerging filtering algorithms in urban scene (%).

Samples	Zhang (2013)	Hu (2014)	Hui (2016)	Shi (2018)	Hui (2018)	Cai (2019)	Wang (2022)	CAP
samp11	18.49	8.31	13.34	11.12	17.1	16.24	17.74	18.04
samp12	5.92	2.58	3.5	7.17	7.14	8.85	5.34	4.43
samp21	4.95	0.95	2.21	6.58	2.55	14.18	4.90	1.87
samp22	14.18	3.23	5.41	14.02	11.47	4.25	8.17	4.47
samp23	12.06	4.42	5.11	17.43	8.86	8.52	8.50	6.63
samp24	20.26	3.8	7.47	13.06	15.96	15.59	8.75	5.05
samp31	2.32	0.9	1.33	3.13	6.82	7.28	4.93	4.17
samp41	20.44	5.91	10.6	10.06	11.45	13.04	7.91	5.93
samp42	3.94	0.73	1.92	1.91	4.13	4.75	3.48	2.50
Avg.	11.40	3.43	5.65	9.39	9.50	10.30	7.75	5.90
Std.	7.32	2.54	4.13	5.14	4.97	4.58	4.21	4.79

3.2. Accuracy Evaluation of CAP in the NRW Experimental Areas

For the experimental data in the NRW region, the results demonstrate that the CAP filtering algorithm outperforms traditional methods, like the PTD algorithm and the CSF algorithm, especially in terms of ground point extraction accuracy and Kappa coefficient. Specifically, the CAP algorithm surpasses both PTD and CSF algorithms across several key performance indicators, particularly in total error (TE) and classification accuracy (Kappa coefficient).

As shown in Table 4, the CAP algorithm achieves an average total error of 2.86%, which is lower than PTD's 4.87% and CSF's 3.46%. This represents a reduction of 2.01%

and 0.60% in absolute error, respectively, and a relative error reduction of 41.27% and 17.34%. The CAP algorithm demonstrates superior precision in classifying ground and object points, with a significant reduction in errors.

Table 4. Different errors and Kappa coefficients computed by three different algorithms (%).

Site	Type I Error			Type II Error			Total Error			Kappa		
	PTD	CSF	CAP	PTD	CSF	CAP	PTD	CSF	CAP	PTD	CSF	CAP
1	10.13	5.86	2.89	3.95	3.46	4.20	7.66	4.89	3.42	84.32	89.94	92.89
2	5.91	6.99	5.53	2.91	2.00	2.31	4.99	5.44	4.53	88.64	87.75	89.71
3	5.16	1.72	0.87	2.12	2.26	2.46	3.44	2.02	1.77	92.98	95.88	96.41
4	6.28	3.75	2.05	1.77	1.82	2.18	4.55	3.00	2.10	90.55	93.74	95.58
5	6.85	4.70	3.94	2.14	1.60	1.75	4.43	3.10	2.81	91.13	93.79	94.37
6	5.00	1.07	0.65	1.25	1.31	1.44	2.90	1.21	1.09	94.10	97.55	97.78
7	5.47	1.80	0.66	5.87	3.87	4.69	5.62	2.61	2.23	88.21	94.50	95.28
8	10.59	8.99	8.11	3.49	3.02	3.24	7.89	6.78	6.31	83.36	85.82	86.76
9	5.99	3.98	2.06	2.58	2.91	3.19	4.32	3.46	2.61	91.36	93.08	94.77
10	7.01	6.25	4.76	2.74	2.23	2.75	5.16	4.58	3.89	89.58	90.74	92.12
11	6.15	2.75	1.37	2.15	2.05	2.32	4.15	2.40	1.85	91.71	95.21	96.30
12	5.97	2.42	1.52	1.82	1.78	1.97	3.37	2.07	1.76	92.48	95.82	96.45
Avg.	6.71	4.19	2.87	2.73	2.36	2.71	4.87	3.46	2.86	89.87	92.82	94.04
Std.	1.81	2.43	2.31	1.24	0.79	0.97	1.56	1.64	1.47	3.30	3.55	3.19

Furthermore, the Kappa coefficient, a key metric for measuring classification consistency and accuracy, further confirms the superiority of the CAP algorithm. The average Kappa coefficient across all 12 sites for the CAP algorithm is 94.04%, higher than PTD's 89.87% and CSF's 92.82%, representing improvements of 4.17% and 1.22%, respectively. These results indicate that the CAP algorithm not only reduces classification errors but also enhances the consistency of the classification results.

From the results of type I and type II errors, the CAP algorithm demonstrates particularly strong performance in reducing type I errors, as shown in Figure 8. According to Table 5, the average error for type I is 2.87%, significantly lower than PTD (6.71%) and CSF (4.19%). Notably, the results at sites 6 (0.65%), 7 (0.66%), and 3 (0.87%) are particularly good, indicating that the CAP algorithm can more accurately distinguish object points.

Table 5. Comparison of time consumption of three different algorithms.

Site	Time Consuming/s			Num/pts	Density/pts/m ²
	PTD	CSF	CAP		
1	4.96	1.45	1.65	96,164	0.49
2	5.23	1.53	1.70	97,583	0.49
3	4.16	0.66	0.81	95,606	0.49
4	4.28	1.01	1.18	94,933	0.50
5	4.37	1.20	1.34	95,433	0.49
6	3.89	0.86	1.01	96,206	0.48
7	4.39	1.21	1.36	93,179	0.49
8	4.21	1.44	1.62	93,934	0.49
9	4.89	0.94	1.11	93,348	0.49
10	4.97	1.31	1.48	94,672	0.49
11	3.82	0.91	1.07	94,280	0.50
12	4.22	0.83	0.99	95,438	0.50
Avg.	4.45	1.11	1.28	95,065	0.49

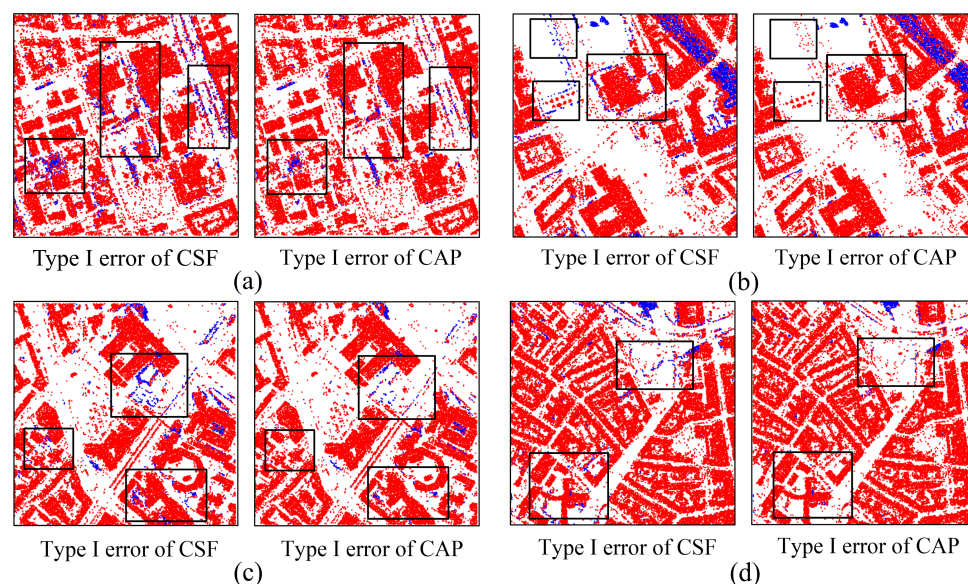


Figure 8. Comparison of type I error between the CSF algorithm and the CAP algorithm. (a–d) respectively represent the 4 corresponding experimental areas in the NRW.

However, the CAP algorithm performs somewhat weaker in reducing type II errors, with an average error of 2.71%. Notably, sites 1 (4.20%) and 7 (4.69%) show larger errors. In comparison, the CSF algorithm performs better in terms of type II errors, with an average error of 2.36%. These findings suggest that while the CAP algorithm excels at minimizing type I errors, its performance on type II errors could be further improved.

In the NRW experimental area, the PTD algorithm performs overall poorly, with the type I error as high as 6.71%, much higher than the CSF and CAP algorithms. The advantage in type II error is not significant either, being similar to CAP but higher than CSF. The total error and Kappa coefficient are lower compared to the CAP algorithm, indicating lower accuracy.

3.3. Time Cost Evaluation of CAP in NRW Experimental Areas

This study conducted a systematic comparative evaluation of the computational efficiency among PTD, CSF, and CAP algorithms. Experiments were performed on sampled NRW experimental areas, with 10 repeated runs on identical hardware configurations to obtain averaged results. As shown in Table 5, the findings demonstrate that the PTD algorithm exhibited an average processing time of 4.45 s, primarily attributed to its multi-iteration TIN model refinement mechanism. The CSF algorithm achieved the shortest processing time (1.11 s) by leveraging holistic simulated terrain surface characteristics. The CAP algorithm adopted a two-stage architecture of “CSF coarse filtering and refined extraction”, combining CSF preprocessing with static TIN modeling and joint-criteria optimization, resulting in a total processing time of 1.28 s–3.48 times faster than PTD. Although CAP introduced a 16.3% time increase compared to CSF due to TIN modeling integration, it significantly reduced computational complexity by replacing PTD’s progressive iterations with single-pass optimization while maintaining classification accuracy.

3.4. Parameter Sensitivity Analysis

Parameters are very important to an algorithm and sometimes can directly affect the quality of the algorithm results. The first step of the CAP algorithm is to extract the ground points of the CSF algorithm, and the parameters of the CSF algorithm involve the RI, ST, GR, and so on, so the best choice of parameters is also the most important thing in the CSF algorithm, and the subsequent angle and distance parameters are also the same. Through

parameter sensitivity analysis, controlling parameter variables, comparing the accuracy of different experimental results, and analyzing the influence of each parameter on filtering performance, the relatively optimal parameter is selected.

As shown in Figure 9, a comparison analysis of total error and time consumption was conducted by randomly selecting 8 out of 12 areas, using different GRs. The line chart illustrates the impact of GR on total error. From the figure, it can be observed that when the GR is in the range of 1–2 m, the total error across the sample areas is relatively low and stable. Particularly, at GR = 1.5 m, the error is minimized for most of the sample areas, and the time consuming is also relatively low. Therefore, GR = 1.5 m is selected as the corresponding parameter for this study.

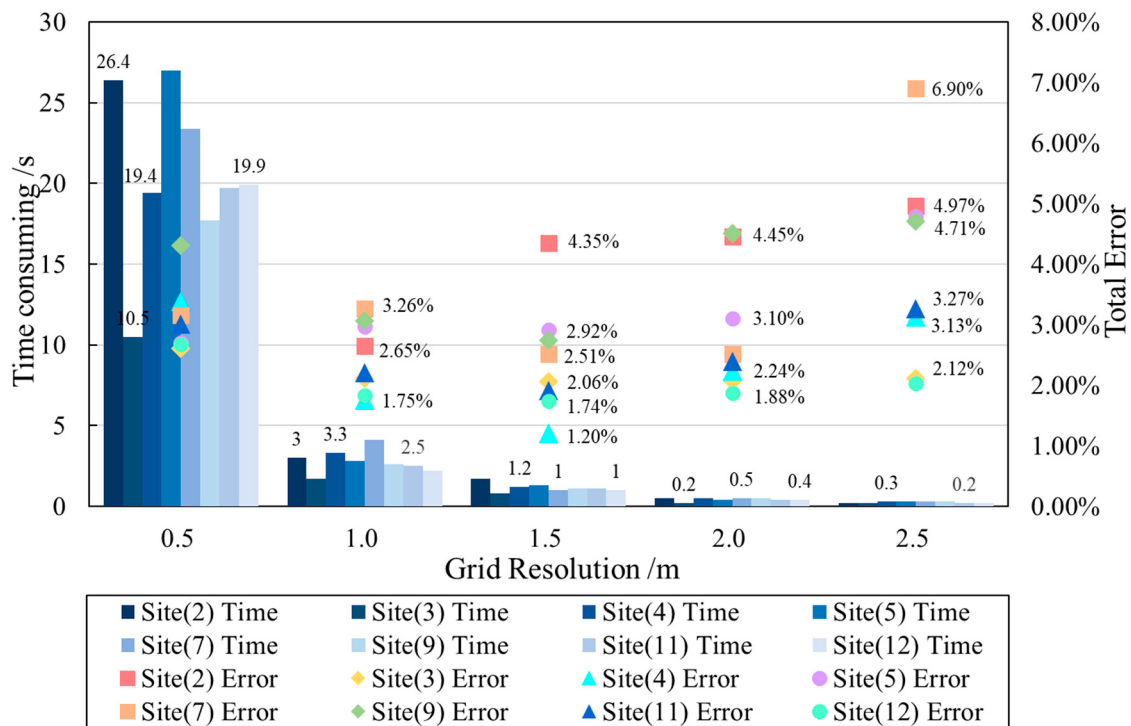


Figure 9. Effect of the GR on the total error of the data.

As shown in Figure 10, at the areas of site 1, site 2, site 8, and site 9, the value of h has a minimal effect on the total error, and the points in the figure are relatively clustered. However, at other areas, such as site 3, site 4, and site 10, different values of h have a significant impact on the total error. Roughly speaking, when $h = 0.5$ m, some areas show the maximum error, especially at site 4. But upon closer analysis, when $h = 0.3$ m, most sites achieve the minimum error. When the remaining parameters are selected optimally, the average total error across the 12 experimental sample areas is reduced by 0.01% compared to when $h = 0.5$ m. The study thus selects $h = 0.3$ m.

Figure 11a shows the effect of RI on the error of the experimental sample areas. From this figure, it can be seen that RI has little impact on the total error in certain areas, such as site 1, site 2, site 5, and site 8. Regardless of the RI value, the total error remains relatively stable. However, there are significant differences in some areas, such as site 4, site 7, and site 9. When RI = 1, the total error in these areas is smaller. When RI = 2 and 3, the total error across the 12 areas is similar, and the differences are not significant. Figure 11b shows the effect of ST on the experimental sample areas. Whether ST is set or not has a considerable impact on the total error of the areas. When ST = True, the total error in all areas, except site 2, is smaller than when ST = False. Based on the above analysis, this study selects RI = 1 and ST = True for the NRW region areas.

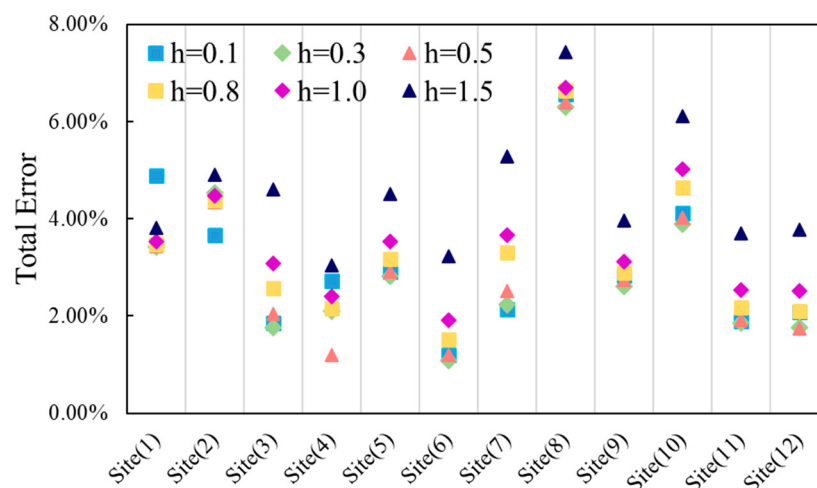


Figure 10. The threshold of the ground pint classification (h) on the error of the experimental sample area.

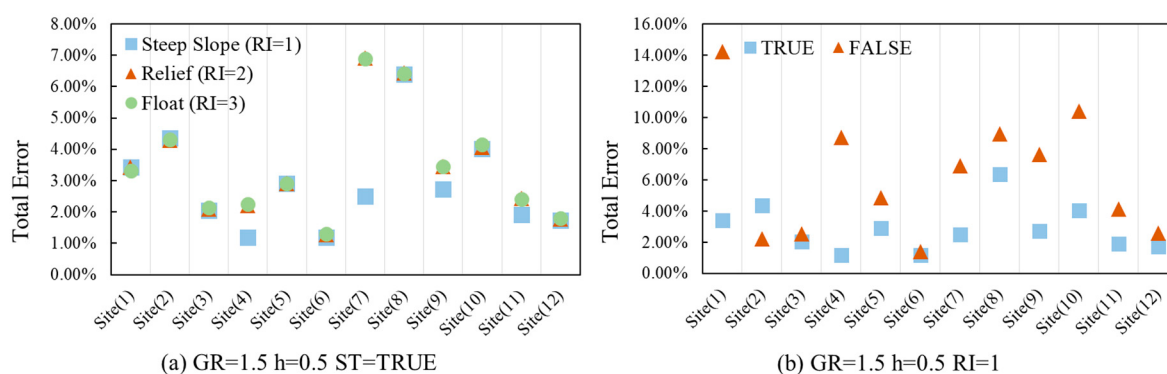


Figure 11. RI and ST on the error of the experimental sample area.

From the Figure 12, it can be observed that the changes in type I and type II errors exhibit an inverse trend. As the angle threshold decreases, the type II error decreases while the type I error increases. Under a fixed angle, the type I error increases as the distance threshold decreases, while the type II error and total error gradually decrease as the distance decreases. The filtering effect is most ideal when the distance threshold is in a smaller range (0.5–1.0 m). Therefore, the middle value of 0.8 m was adopted as the distance threshold of the experiment. When the distance is fixed, as the angle decreases, the total error shows a decreasing trend with the reduction in distance. Notably, when the angle is set to 6° , the total error in the experimental area remains stable at approximately 2.38%, indicating that 6° is a relatively stable choice under the experimental conditions.

A combination of larger angles and longer distances leads to an increase in type II error, which in turn affects the total error. For example, in the combination of 30° and 1.5 m, although the type I error is small (0.28%), the type II error is large (6.94%), resulting in a total error of 3.53%, which is significantly higher than other angle combinations.

The angle threshold selected in the experiment was 6° because this angle showed relatively stable characteristics under the conditions of this experiment and the total error is small at 6° .

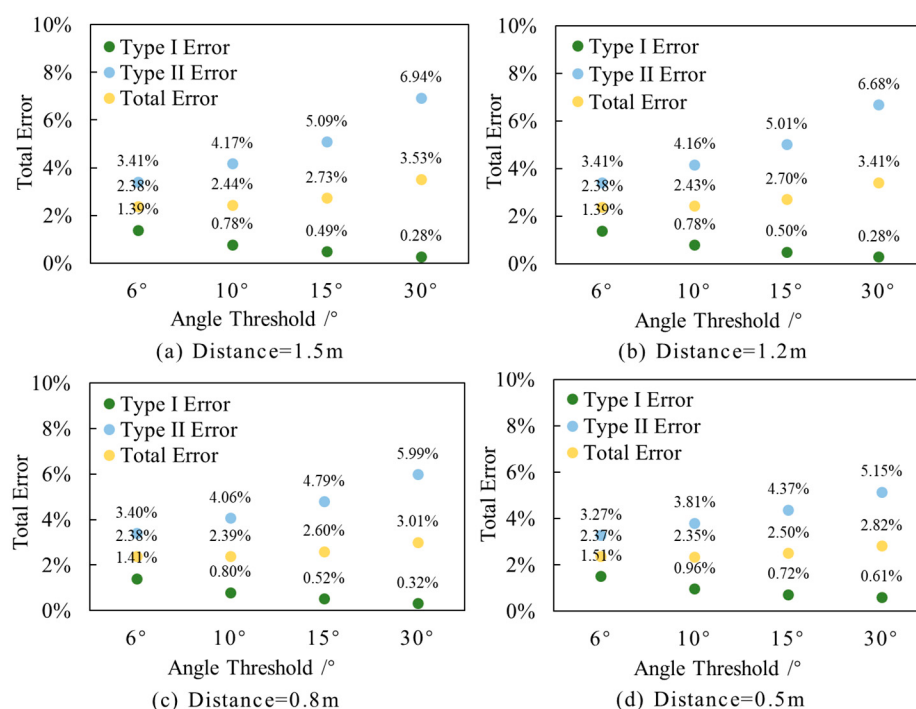


Figure 12. Effect of angle threshold on three types of errors under a constant distance threshold.

4. Discussion

4.1. Algorithmic Synergy and Performance Improvements

The innovation of CAP framework lies in its seed point optimization strategy: by replacing the random initial seeds of traditional PTD algorithm with accurate ground points preprocessed by CSF, the instability problem of triangulation network construction caused by the sparsity of initial seeds in complex urban terrain is solved. CSF's cloth simulation mechanism generates morphologically consistent and spatially dense seed point sets (e.g., roads, slope continuous features), which significantly enhance the robustness of triangulation in irregular regions (building edges, vegetation areas). This strategy combines physically driven seed screening with subsequent TIN construction, giving priority to ensuring accuracy and reducing the need for iterative correction. Furthermore, the angular distance judgment mechanism addresses CSF's limitations in handling irregular geometries (e.g., building edges, road discontinuities). Compared with the independent implementation of CSF and PTD, the accuracy of type I error, total error and Kappa coefficient index are correspondingly improved.

4.2. Cause Analysis of Type II Error Anomaly in CAP Algorithm

In this study of filtering algorithms, the CAP algorithm demonstrates a technical dependency on the CSF algorithm. Specifically, CAP constructs the TIN using the initial ground points identified by CSF, then performs secondary filtering on unclassified points. While the CSF-generated initial ground point category achieves substantial accuracy, residual ground point errors persist. To address this, CAP incorporates the PTD concept, implementing multi-threshold iterations for refined ground extraction.

Regarding performance metrics, CAP generally surpasses CSF in overall accuracy, yet their relationship is not strictly hierarchical. As shown in Figure 13's scatter distribution, CAP exhibits higher type II errors. This stems from its TIN-based geometric constraints: when unclassified points (particularly those adhering to terrain surfaces, like low vegetation or road barriers) satisfy predefined angle and distance thresholds, they risk erroneous ground classification. The limitation becomes pronounced in complex urban environments

containing scattered objects, where geometric models struggle to distinguish terrain from surface-attached features. While CAP does an excellent job of optimizing CSF results by adding threshold decision processing, its performance boundaries ultimately depend on the accuracy of the initial ground model. Such algorithm cascading enhances the overall precision but simultaneously inherits and amplifies predecessor limitations, highlighting challenges in applying pure geometric filtering to complex environments. Future research can go beyond the geometry-driven paradigm and integrate multimodal data (e.g., spectral texture, temporal motion features) and dynamic contextual reasoning to achieve a transition from “geometric filtering” to “joint semantic–geometric understanding”.

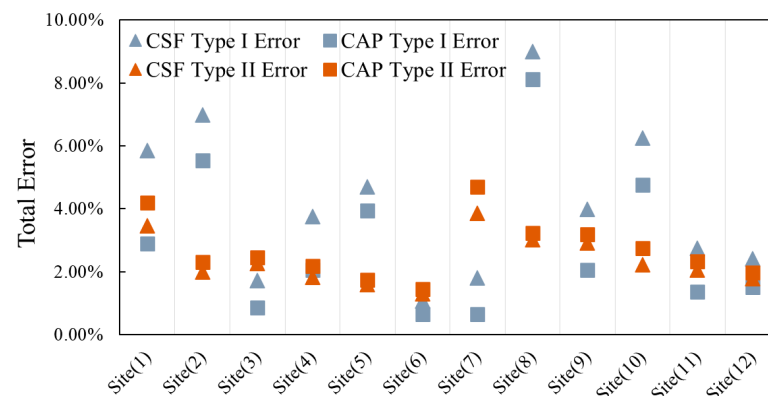


Figure 13. The type I and type II errors of the CSF and CAP algorithms.

4.3. Uneven Density in TIN Model Construction Using CAP Algorithm

There are some challenges when using CSF algorithm to extract ground points from original urban scene point cloud data and construct TIN model, especially in densely built urban areas. As shown in Figure 14, when filtering object points, the CSF algorithm tends to roughly eliminate the object point cloud data, such as buildings, resulting in the TIN triangulated network surface constructed in these areas is too rough and the area is too large. On the contrary, in the non-building area, the ground points are more average and dense, and then a more refined TIN model is constructed. This phenomenon can lead to obvious differences in the construction effects of TIN models within the same region. In the vicinity of buildings, due to the lack of object points, the accuracy of the model is low, showing a rough triangular network structure. However, in non-building areas, due to the dense point cloud data and accurate ground point extraction, the model is relatively fine and has high accuracy, and this difference may affect the overall accuracy of the whole algorithm.

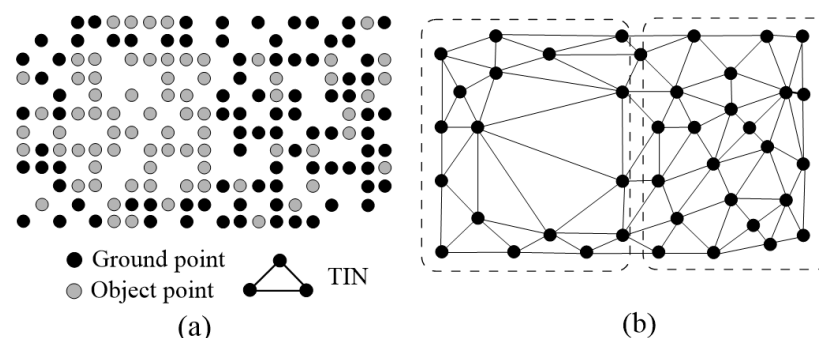


Figure 14. The original point cloud data and the construction of the TIN model filtered by CSF. (a) Original point cloud data classification containing object points, such as buildings, trees, and vehicles, and ground points. (b) TIN model constructed from ground points obtained using the CSF algorithm.

In the future, a local–global joint optimization strategy can be designed to introduce super-voxel segmentation in built-up areas to refine the granularity of terrain reconstruction, and combine dynamic terrain curvature constraints to suppress excessive smoothing in non-built-up areas. In addition, the incremental TIN encryption technology was explored, and the triangle patches were added and deleted adaptively based on point cloud density and geometric complexity to achieve the equalization of model accuracy in the whole region.

4.4. Future Research Directions

Considering the diversity of urban environments and their potential impact on algorithm performance, urban studies in special scenarios and verification of parameter generalization ability in different environments are crucial. Different geographical conditions can significantly affect the applicability and robustness of the algorithm. Therefore, exploring the performance of the algorithm in different urban environments is crucial to enhance its generalization ability. At the same time, verifying the generalization ability of the parameters in different scenarios helps to ensure that the algorithm maintains stability and accuracy in dynamic practical applications. Future research can focus on the applicability of the algorithm in areas with unique terrain conditions, such as mountain cities and coastal cities, to further evaluate its performance in various urban environments. In addition, since the current research has not fully verified the generalization ability of the parameters in different scenarios, cross-scenario experiments can be carried out in the future to analyze the impact of parameter changes on the performance of the algorithm in different environments, and explore ways to improve the generalization ability of the algorithm.

5. Conclusions

This paper proposes an improved algorithm framework for urban point cloud filtering, which combines PTD and CSF algorithm for structure reconstruction, so that the advantages of the two algorithms are complementary, and the balance between sparse—dense nodes and the complexity of multi-scale structure in urban environment point cloud data are solved, so as to improve the accuracy of point cloud filtering in its environment.

The paper compares its filtering algorithm with 14 others using the ISPRS dataset. It achieves a low total error of 1.87% in sample 21, but a higher error of 18.04% in sample 11, indicating room for improvement in challenging terrains. It outperforms many emerging algorithms, showing good reliability and applicability.

On the urban point cloud data of NRW, the CAP algorithm performs better than the PTD and CSF algorithms in terms of ground point extraction accuracy, classification agreement, and type I error. The average total error of the proposed algorithm is 2.86%, which is lower than that of PTD algorithm (4.87%) and CSF algorithm (3.46%). It effectively reduces the relative error, indicating that it has higher ground point classification accuracy in complex urban environments. The Kappa coefficient of the CAP algorithm is 94.04%, which shows that its classification consistency is significantly improved, and it is more robust to deal with point cloud data in urban environments. In addition, the CAP algorithm significantly reduces the type I error to 2.86%, although the performance on the type II error is relatively weak (3.71%), but compared with the other two algorithms, the TE is improved. The CAP algorithm achieves a 1.28 s processing time via a two-stage architecture, operating 3.48 times faster than PTD. Despite a 16.3% time increase compared to CSF, it replaces iterative computations with single-pass optimization, significantly improving efficiency while maintaining classification accuracy.

In general, the CAP algorithm provides a promising solution for high-precision ground point extraction, especially in complex urban landscapes. Compared with traditional methods, the accuracy of CAP algorithm shows stronger advantages. Future research

can focus on further optimizing the algorithm to adapt it to larger datasets and more challenging urban environments and exploring automated processing of parameterization to enhance its applicability.

Author Contributions: Data curation, D.Y., L.Z. and R.Y.; Formal analysis, G.L.; Funding acquisition, L.Z. and R.Y.; Methodology, G.L. and X.C.; Project administration, L.Z. and R.Y.; Software, G.L.; Supervision, L.Z. and R.Y.; Visualization, X.C. and D.Y.; Writing—original draft, G.L. and X.C.; Writing—review & editing, D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Natural Science Foundation of China (52274169 and 52174160) and priority projects for the “Science and Technology for the Development of Inner Mongolia” initiative in 2023 (ZD20232304).

Data Availability Statement: The original data presented in the study are openly available in open data at <https://www.itc.nl/isprs/wgIII-3/filtertest/downloadsites/> (accessed on 12 August 2024); https://www.opengeodata.nrw.de/produkte/geobasis/hm/3dm_1_las/3dm_1_las/ (accessed on 12 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Awrangjeb, M.; Lu, G.; Fraser, C. Automatic building extraction from LiDAR data covering complex urban scenes. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2014**, *40*, 25–32. [CrossRef]
2. Guo, L.; Deng, X.; Liu, Y.; He, H.; Lin, H.; Qiu, G.; Yang, W. Extraction of Dense Urban Buildings From Photogrammetric and LiDAR Point Clouds. *IEEE Access* **2021**, *9*, 111823–111832. [CrossRef]
3. Sevgen, S.C.; Karsli, F. Automatic ground extraction for urban areas from airborne LiDAR data. *Turk. J. Eng.* **2020**, *4*, 113–122. [CrossRef]
4. Balado Frías, J.; González Rodríguez, M.E.; Verbree, E.; Díaz Vilariño, L.; Lorenzo Cimadevila, H.R. Automatic detection and characterization of ground occlusions in urban point clouds from mobile laser scanning data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *VI-4/W1-2020*, 503–508. [CrossRef]
5. Dogan, Y.; Ok, A.Ö. A feature-based deep learning approach for the extraction of ground points from 3D point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *XLVIII-M-1-2023*, 503–508. [CrossRef]
6. Sun, H.; Liu, J.; Zhang, L.; Sun, H. A Laser Inertial SLAM System with High Robustness and Low Drift in Complex Environments. In Proceedings of the 2024 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2024. [CrossRef]
7. Dell, F.; Acqua, G.; Gamba, P.; Mainardi, A. Digital Terrain Models in Dense Urban Areas. *Int. Arch. Photogramm. Remote Sens.* **2001**, *33*, 935–942. Available online: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b2806e72f8c77d140141ecff2537420963de66b1> (accessed on 14 August 2024).
8. Ribero, M.; Heath, R.W.; Vikalo, H.; Chizhik, D.; Valenzuela, R.A. Deep Learning Propagation Models over Irregular Terrain. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019. [CrossRef]
9. Yan, Y.; Zhang, J.; Li, J. Review of ground filtering algorithms for airborne lidar point cloud data. *Beijing Surv. Mapp.* **2024**, *38*, 1513–1520. [CrossRef]
10. Axelsson, P. DEM generation from laser scanner data using adaptive TIN models. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2000**, *33*, 110–117. Available online: <https://api.semanticscholar.org/CorpusID:59795495> (accessed on 7 February 2025).
11. Zhang, J.; Lin, X. Filtering Airborne LiDAR Data by Embedding Smoothness-Constrained Segmentation in Progressive TIN Densification. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 44–59. [CrossRef]
12. Cai, S.; Zhang, W.; Liang, X.; Wan, P.; Qi, J.; Yu, S.; Yan, G.; Shao, J. Filtering Airborne LiDAR Data Through Complementary Cloth Simulation and Progressive TIN Densification Filters. *Remote Sens.* **2019**, *11*, 1037. [CrossRef]
13. Kraus, K.; Pfeifer, N. Advanced DTM generation from LiDAR data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2001**, *34*, 23–30. Available online: https://www.researchgate.net/publication/51997520_Advanced_DTM_generation_from_LIDAR_data (accessed on 7 February 2025).
14. Hu, H.; Ding, Y.; Zhu, Q.; Wu, B.; Lin, H.; Du, Z.; Zhang, Y.; Zhang, Y. An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 98–111. [CrossRef]

15. Chen, Q.; Gong, P.; Baldocchi, D.; Xie, G. Filtering Airborne Laser Scanning Data with Morphological Methods. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 175–185. [\[CrossRef\]](#)
16. Pingel, T.J.; Clarke, K.C.; McBride, W.A. An Improved Simple Morphological Filter for the Terrain Classification of Airborne LiDAR Data. *ISPRS J. Photogram. Remote Sens.* **2013**, *77*, 21–30. [\[CrossRef\]](#)
17. Wu, W. A Connected Sets Detection Morphological Filter for Airborne LiDAR DTM Extraction under Urban Area. In Proceedings of the 2023 8th International Conference on Computer and Communication Systems (ICCCS), Guangzhou, China, 21–24 April 2023. [\[CrossRef\]](#)
18. Sithole, G.; Vosselman, G. Filtering of airborne laser scanner data based on segmented point clouds. *Int. Arch. Photogram. Rem. Sens. Spat. Inform. Sci.* **2005**, *36 Pt 3*, 66–71. Available online: <https://api.semanticscholar.org/CorpusID:13413535> (accessed on 15 July 2024).
19. Lin, X.; Zhang, J. Segmentation-Based Filtering of Airborne LiDAR Point Clouds by Progressive Densification of Terrain Segments. *Remote Sens.* **2014**, *6*, 1294–1326. [\[CrossRef\]](#)
20. Chen, F.; Xie, F.; Sun, L.; Gu, Y.; Zhang, Z.; Chen, J.; Zhang, J.; Yi, M. Point cloud segmentation algorithm based on improved Euclidean clustering. *IEEE Access* **2024**, *12*, 152959–152971. [\[CrossRef\]](#)
21. Zhao, F.; Huang, H.; Xiao, N.; Yu, J.; Geng, G. A point cloud segmentation algorithm based on multi-feature training and weighted random forest. *Meas. Sci. Technol.* **2024**, *36*, 015407. [\[CrossRef\]](#)
22. Salah, M.; Farhan, M.; Basha, A. Filtering of 3D point clouds using maximum likelihood algorithm. *Discov. Appl. Sci.* **2024**, *6*, 419. [\[CrossRef\]](#)
23. Lian, Z.; Gu, Y.; You, K.; Xie, X.; Qiu, G. An adaptive multi-scale point cloud filtering method for feature information retention. *Opt. Lasers Eng.* **2024**, *177*, 108144. [\[CrossRef\]](#)
24. Qi, C.; Wang, X.; Su, D.; Guo, Y.; Yang, F. Comparison and analysis of ground seed detectors and interpolation methods in airborne LiDAR filtering. *Egypt. J. Remote Sens. Space Sci.* **2023**, *26*, 1009–1019. [\[CrossRef\]](#)
25. Li, H.; Li, D.; Huang, X. An Improved filtering algorithm for LiDAR Point Cloud Based on Progressive Encryption Triangulation. *Sci. Surv. Mapp.* **2009**, *34*, 39–40. Available online: <https://www.cnki.com.cn/Article/CJFDTotal-CHKD200903012.htm> (accessed on 24 August 2024).
26. Sui, L.; Zhang, Y.; Zhang, S. Filtering of airborne LiDAR point cloud Data based on Progressive triangulation. *J. Wuhan Univ. (Sci. Inf. Ed.)* **2011**, *36*, 1159–1163. Available online: <http://ch.whu.edu.cn/article/id/688> (accessed on 11 August 2024).
27. Li, H.; Ye, C.; Guo, Z.; Wei, R.; Wang, L.; Li, J. A Fast Progressive TIN Densification Filtering Algorithm for Airborne LiDAR Data Using Adjacent Surface Information. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12492–12503. [\[CrossRef\]](#)
28. Shi, X.; Ma, H.; Chen, Y.; Zhang, L.; Zhou, W. A parameter-free progressive TIN densification filtering algorithm for LiDAR point clouds. *Int. J. Remote Sens.* **2018**, *39*, 6969–6982. [\[CrossRef\]](#)
29. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 501. [\[CrossRef\]](#)
30. Li, Y.; Li, C.; Wang, S. An Improved Cloth Simulation Filtering Method Considering Terrain Characteristics. *Remote Sens. Technol. Appl.* **2019**, *34*, 1261–1268. Available online: <http://www.rsta.ac.cn/CN/10.11873/j.issn.1004-0323.2019.6.1261> (accessed on 21 September 2024).
31. Cai, S.; Yu, S.; Hui, Z.; Tang, Z. ICSF: An Improved Cloth Simulation Filtering Algorithm for Airborne LiDAR Data Based on Morphological Operations. *Forests* **2023**, *14*, 1520. [\[CrossRef\]](#)
32. Yang, A.; Wu, Z.; Yang, F. An Automatic Filtering Algorithm of Multi-beam Bathymetry Based on Bidirectional Cloth Simulation. *Geomat. Inf. Sci. Wuhan Univ.* **2012**, *47*, 517–525. [\[CrossRef\]](#)
33. Xue, D.; Cheng, Y.; Shi, X. Point cloud classification algorithm based on cloth filtering and improved random forest. *Laser Optoelectron. Prog.* **2020**, *57*, 192–200. Available online: <https://researching.cn/ArticlePdf/m00002/2020/57/22/221017.pdf> (accessed on 14 September 2024).
34. Zheng, J.; Xiang, M.; Zhang, T.; Zhou, J. An Improved Adaptive Grid-Based Progressive Triangulated Irregular Network Densification Algorithm for Filtering Airborne LiDAR Data. *Remote Sens.* **2024**, *16*, 3846. [\[CrossRef\]](#)
35. Niedzwiedzki, J.; Lipinski, P.; Podsedkowski, L. IDTMM: Incremental Direct Triangle Mesh Mapping. *IEEE Robot. Autom. Lett.* **2023**, *8*, 5416–5423. [\[CrossRef\]](#)
36. Li, Y.; Chen, Q.; Li, C.; Bu, P. Improvement of CSF Based on a Wide Range of Urban Complex Scenes. In *Lidar Remote Sensing for Environmental Monitoring XVI*; SPIE: Bellingham, WA, USA, 2018; Volume 10779, p. 1077911. [\[CrossRef\]](#)
37. District Government of Cologne. Available online: <https://www.bezreg-koeln.nrw.de> (accessed on 29 March 2025).
38. Ren, D.W. UAV Oblique Image Dense Matching and Two Kinds of Point Cloud Fusion Technology Combined with Ground LiDAR Point Cloud. Master's Thesis, Wuhan University, Wuhan, China, 2022.
39. Zhi, Z.; Chang, B.; Li, Y. P-CSF: Polar coordinate cloth simulation filtering algorithm for multi-type tunnel point clouds. *Tunn. Undergr. Space Technol.* **2025**, *155*, 106144. [\[CrossRef\]](#)

40. Provot, X. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour. In *Graphics Interface*; Canadian Information Processing Society: Mississauga, ON, Canada, 1995.
41. Dong, Y.; Cui, X.; Zhang, L.; Ai, H. An Improved Progressive TIN Densification Filtering Method Considering the Density and Standard Variance of Point Clouds. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 409. [[CrossRef](#)]
42. Zhao, X.; Guo, Q.; Su, Y.; Xue, B. Improved Progressive TIN Densification Filtering Algorithm for Airborne LiDAR Data in Forested Areas. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 79–91. [[CrossRef](#)]
43. Hui, Z.; Hu, Y.; Yeveyo, Y.; Yu, X. An Improved Morphological Algorithm for Filtering Airborne LiDAR Point Cloud Based on Multi-Level Kriging Interpolation. *Remote Sens.* **2016**, *8*, 35. [[CrossRef](#)]
44. Hui, Z.; Li, D.; Jin, S.; Ziggah, Y.; Wang, L.; Hu, Y. Automatic DTM extraction from airborne LiDAR based on expectation maximization. *Opt. Laser Technol.* **2018**, *112*, 43–55. [[CrossRef](#)]
45. Wang, W.; Li, Z.; Fu, Y.; He, H.; Xiong, F. A Multi-scale Adaptive Slope Filtering Algorithm for Point Cloud. *Geomat. Inf. Sci. Wuhan Univ.* **2022**, *47*, 438–446. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.