

A data-driven morphological filtering algorithm for digital terrain model generation from airborne LiDAR data

Bingxiao Wu ^{a,b,c,*}, Xingxing Zhou ^{a,b,**}, Junhong Zhao ^{a,b}, Wuming Zhang ^c, Guang Zheng ^d

^a Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences, Guangzhou, 510640, China

^b Guangdong Engineering Technology Research Center of Smart Facility Agriculture, Guangzhou, 510640, China

^c School of Geospatial Engineering and Science, Sun Yat-Sen University, Zhuhai, 519082, China

^d International Institute for Earth System Science, Nanjing University, Nanjing, 210023, China

ARTICLE INFO

Keywords:

Ground filtering algorithm
Topographic features
Digital terrain model
Airborne point cloud
 k -dimensional tree

ABSTRACT

Ground filtering algorithms (GFs) are widely used in point cloud processing to generate digital terrain models. Existing GFs typically rely on rule-based or machine learning approaches to separate ground and non-ground points within an airborne point cloud. However, they often struggle to accurately extract ground points in scenarios containing mountains and heterogeneous buildings. To enhance the accuracy and robustness of ground filtering for airborne point clouds, we propose a data-driven morphological filtering algorithm (DMF). DMF begins by identifying near-ground voxel centroids after voxelizing the input point clouds. Next, a digital elevation model is constructed based on the elevation information of these near-ground voxel centroids. A composite morphological filter is then designed to identify ground and non-ground patches within the digital elevation model before labeling their inner near-ground voxel centroids as GF-support nodes. The composite morphological filter is used to recognize non-ground areas with incomplete edge structures depicted in the input point cloud and to correct misclassified areas. Finally, a bidirectional k -dimensional tree search engine is built between the GF-support nodes and the input point cloud to separate ground and non-ground points. Experimental results show that DMF achieves ground filtering accuracy with an average F-score greater than 0.88, demonstrating robustness in generating digital terrain models across various test scenarios. Furthermore, the intermediate outputs of DMF enable instance segmentation of artificial objects in airborne point clouds. The code for DMF will be shared on GitHub (<https://github.com/wbx1727031/DMF>).

1. Introduction

With the development of airborne laser scanning (ALS) technology, the return signal from multi-epoch laser pulses reflected from objects on the ground can be captured and decoded as discrete point clouds to depict the 3D structure features of scanned targets and their surface morphological features (Eitel et al., 2016). The ALS data shows under-story topographic features and canopy vertical structure, essential for topographic mapping and forest ecology research (Liang et al., 2019; Meng et al., 2010; Wei-Lwun et al., 2009). The distribution of point clouds depicts slope and aspect of valleys and ridges, aiding in planning operational paths for agricultural machinery, finding suitable building locations, and selecting ideal sites for orchard planting in mountainous

scenes (Bai et al., 2023; Du et al., 2023; Orhan, 2021; Zhao et al., 2024). Due to the mixing of ground and non-ground points in the ALS datasets, it is challenging to separate these two types of points to obtain topographic features in the scanned regions (Chen et al., 2016; Montealegre et al., 2015; Zhao et al., 2018).

The tools used to extract ground points from an ALS dataset are known as ground filtering algorithms (GFs) (Klápsté et al., 2020). GFs are designed based on the principle that laser pulses within common-use bandwidth cannot penetrate soil layers or opaque building surfaces. However, GFs are often hindered by ground point omissions in dense forests and building-covered areas due to the limited penetration of laser pulses through high-density vegetation (Meng et al., 2010; Montealegre et al., 2015; Qin et al., 2023a; Razak et al., 2011). Low scanning angles

This article is part of a special issue entitled: Topographic Mapping from Space published in ISPRS Open Journal of Photogrammetry and Remote Sensing.

* Corresponding author. Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences, Guangzhou, 510640, China.

** Corresponding author. Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences, Guangzhou, 510640, China.

E-mail addresses: wbx19911026@outlook.com (B. Wu), zhouxingxing@gdaas.cn (X. Zhou).

and sparse flight paths during ALS data collection result in an incomplete depiction of building sides and ground near eaves, complicating the GF process (Qin et al., 2017; Razak et al., 2011). Additionally, sensor and environmental factors introduce low-value noise points below the ground and high-value noise points near the sensor's flying height, further affecting GF performance (Eitel et al., 2016; Klápková et al., 2020).

Existing GFs can be categorized into rule-based and machine-learning types (Gharineiat et al., 2022; Qin et al., 2023a). Initial rule-based GFs often rely on fixed-height-difference thresholds to distinguish ground from non-ground points, which works well in flat terrains but less so in hilly landscapes, building areas, and dense forests (Klápková et al., 2020; Montealegre et al., 2015). Some GFs use spline or nonlinear fitting models to simulate uneven ground surfaces (Maguya et al., 2013; Su et al., 2015), while they are sensitive to the surface fitting model selection and noise points (Bulatov et al., 2021; Chen et al., 2021). Improved GFs address these issues by setting data-driven parameters based on local point density or height differences to improve their robustness (Bulatov et al., 2021; Meng et al., 2010).

The other type of GF analyze local morphological features of point clouds to filter ground points, such as elevation skewness, slope, or surface curvature (Bartels and Wei, 2010; Evans and Hudak, 2007; Mongus and Žalik, 2012). They first divide a point cloud into grid cells to capture the lowest point within each cell before constructing a near-ground layer or projecting the point cloud into a digital elevation model (Keqi et al., 2003; Li et al., 2013). Point clusters from non-ground objects and ground can be separated by performing morphological filtering on the near-ground node layer or digital elevation model (Pingel et al., 2013). Existing studies also construct the triangulation network of near-ground nodes and iteratively densify it to separate two types of points from ALS datasets (Cao et al., 2024; Nie et al., 2017; Zhao et al., 2016), which are superior at extracting ground points scanned from mountains and undulating terrains (Cai et al., 2019; Razak et al., 2011). A novel GF builds a multiscale triangulation network and a modified 3D alpha shape to improve its accuracy (Cao et al., 2024).

Cloth simulation filter (CSF), a popular GF, assumes that a cloth-like grid covers the elevation-inverted ALS data (Zhang et al., 2016). The “cloth” falls due to virtual gravity, and ground points can be filtered by calculating height difference between each “cloth” node and its nearest point in the ALS data. Improved CSF focuses on increasing the integrity of ground points extraction on slopes by optimizing seed selection and combining it with a triangulation-network-based GF (Chen et al., 2016; Zhang et al., 2016). Besides, multi-resolution hierarchical cloth simulation and bidirectional internal force operations help smooth spikes in the “cloth” to improve its accuracy (Cai et al., 2024).

Although rule-based GFs are interpretable and easy to use, their performance relies on distinctive point distribution feature selection or threshold settings for separating ground and non-ground points, which may affect their robustness (Cai et al., 2024; Hu and Yuan, 2016). With the development of machine learning algorithms for 3D object segmentation, GFs built based on deep learning (DL) architecture have recently gained attention (Jin et al., 2020; Li et al., 2022). DL-based GFs can be trained with the model architecture of multilayer perceptron (Gharineiat et al., 2022), convolutional neural networks (CNNs) (Liu et al., 2019; Qi et al., 2017; Thomas et al., 2019), graph neural networks (Wang et al., 2019), and transformer model with self-attention mechanism (Cheng et al., 2023; Guo et al., 2021). These DL models always learn features from the 3D coordinate matrix of local point clusters to capture their distribution patterns in diverse scales (Nurunnabi et al., 2021; Wu et al., 2024).

DL-based GFs depend on the encoder layers' ability to perceive and extract local point distribution patterns before transitioning from low-to-high-level attributes for semantic judgment (Dai et al., 2024; Qin et al., 2023b). However, point clouds are disordered and unstructured, making it difficult to design reliable feature extractors for local point clusters (Qi et al., 2017). To address this, some DL models project point clouds into images, extract local surface normal or curvatures, construct

topological graphs for discrete points, or voxelize point clouds to capture useful features for supporting 3D object segmentation (Han et al., 2023; Nurunnabi et al., 2021; Wang et al., 2019). Novel model also use self-attention blocks and multiscale feature fusion mechanisms to enhance the 3D segmentation accuracy (An et al., 2024; Qi et al., 2017; Wu et al., 2024).

The precision and robustness of DL models depend on the quality, diversity, and volume of training datasets (Qin et al., 2023a). Despite numerous model architectures and pre-trained models, they are not specifically designed to achieve ground filtering (Xu et al., 2023). DL model requires setting the cluster number, point number or searching radius when extracting features of local point cluster for model training, which impacts their robustness (Liu et al., 2019; Nurunnabi et al., 2021; Thomas et al., 2019). Additionally, uninterpretable ground filtering processes limit their reliability when applied to scenarios with limited training samples (Qin et al., 2023b).

Given the ground filtering accuracy and avoidance of training cost, users are inclined to use rule-based GFs. Designing an accurate, easy-to-use, and robust GF suitable for complex scenes is challenging but holds significant value (Bulatov et al., 2021; Li et al., 2021; Mongus and Žalik, 2012). To achieve this goal, we developed a data-driven morphological filtering algorithm (DMF), where parameters can be set based on statistical information about point distances and the proposed bidirectional k -dimensional tree (KD-tree) search engine. DMF not only accurately extracts ground points across various scenarios but also provides intermediate outputs that assist in the semantic segmentation of artificial objects in point clouds, a feature unique to most GFs. The DMF processes are described in the following section.

2. Methodology

After voxelizing the input point cloud, DMF extracts centroids of near-ground voxels as a node layer to depict the draft undulating trend of the ground. Next, a composite morphological filter is used to separate nodes distributed within regions covered by non-ground objects or ground. Finally, DMF extracts ground points using a bidirectional k -dimensional tree (KD-tree) search engine built between the node layer and the input point cloud. The flowchart of DMF is shown as follows (Fig. 1).

2.1. Voxelization

DMF first voxelizes the input point cloud. Randomly selecting 1 % of points from the input point cloud. Next, the average Euclidean distance between each chosen point and its nearest neighbor is calculated. To maintain structure details of an input point cloud in its voxelization results, we determined W as.

$$W = \lceil v_0 \cdot \overline{D_m} \rceil. \quad (1)$$

According to the ablation test, v_0 is set 2. We then extracted centroid of non-empty voxels to reduce data volume while retaining coarse distribution pattern of the point cloud (Fig. 2C).

2.2. Near-ground voxel centroids extraction

Given that laser pulse cannot pass through the ground, we can extract centroids from near-ground voxels (ND_{CEs}) (Fig. 2D). However, laser pulses in the green, near-infrared, and short-wave infrared bands cannot penetrate buildings and provide limited penetration of dense forest canopies (Chen et al., 2016; Eitel et al., 2016; Liang et al., 2019). Thus, some ND_{CEs} are from rooftops and canopy surfaces. In addition, ND_{CEs} with abnormal elevations are caused by noise in the input point cloud. Removing noise and misclassified ND_{CEs} is essential for optimizing near-ground voxel centroids extraction.

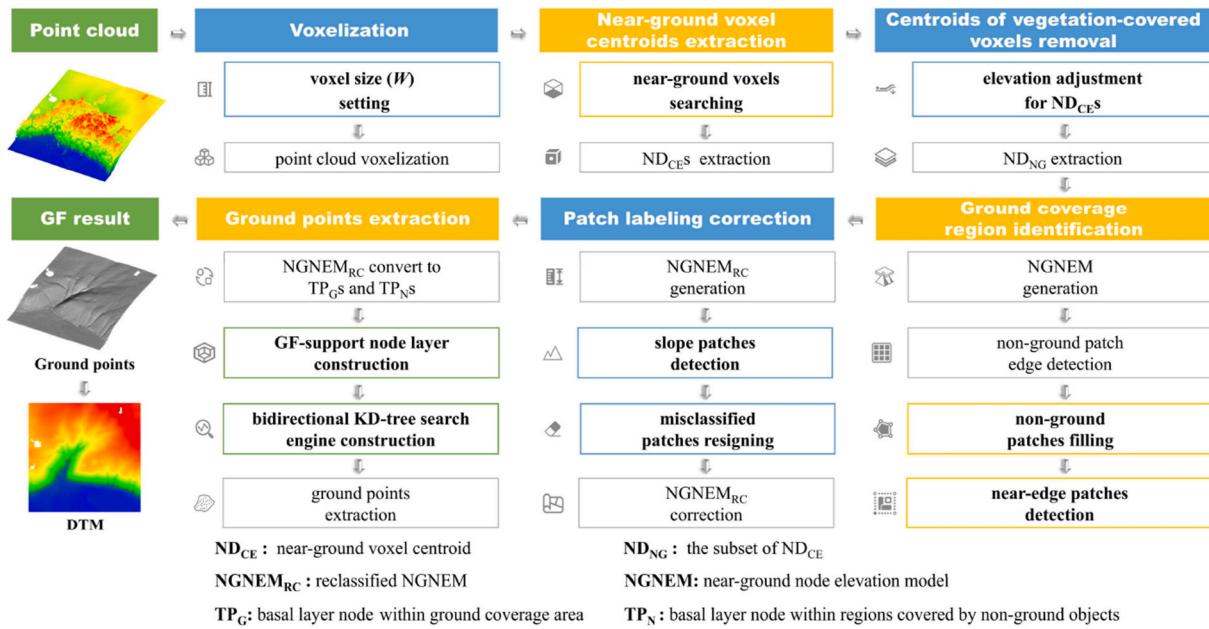


Fig. 1. Flowchart of the proposed data-driven morphological filtering algorithm to generate a digital terrain model for the input point cloud. The bold labels in the colored blocks are the main steps.

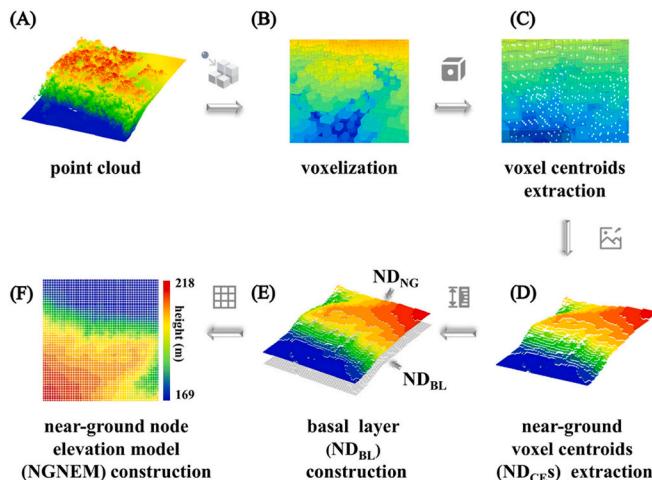


Fig. 2. The near-ground node elevation model construction for an input point cloud.

2.3. Centroids of vegetation-covered voxels removal

After extracting ND_{CEs} from the input point cloud, we iteratively decreased their elevation to construct a basal point layer below the input point cloud to remove ND_{CEs} from vegetation-covered voxels. First, a two-dimensional KD-tree for the ND_{CEs} is constructed, with the search radius set to obtain local centroid clusters C_p shown in cylindrical distribution along the z-axis of each ND_{CE}. Next, we compared the elevation of the other ND_{CEs} in C_p with that of the target node G. If some ND_{CEs} in C_p were lower than G, we calculated their median elevation to assign it to G. Suppose the nearest-neighbor ND_{CE} of G in the C_p differs between N_{th} and (N + 1)_{th} iteration, we adjusted its elevation until they are the same. In the results, we obtained a basal layer ND_{BL} below the input point cloud (Fig. 2E). Apart from their elevation, the nodewise (x, y) coordinates of ND_{BL} and ND_{CE} are the same.

For a target node Q in the ND_{BL}, we found its nearest-neighbor ND_{CE} based on the 3D KD-tree search engine to obtain ND_{NG}. Some ND_{CEs} from vegetation-covered voxels and noise points in the input data can be

removed in ND_{NG}. In the next step, we constructed the near-ground node elevation model (NGNEM) in W-m resolution based on ND_{CEs} to separate ground and non-ground regions shown in the NGNEM. We built NGNEM based on ND_{CEs} as voxel centroids from rooftops in ND_{NG} may be incomplete, which bothers the two types of patches classification in the following step.

2.4. Ground coverage region identification

In an ALS point cloud, non-ground objects depicted in the point cloud appear with elevations higher than the nearby ground surface, and the elevation difference between the edges of each non-ground object and its nearby ground surface exceeds W. In contrast, the local elevation variations near cliffs, ridges, and steep slopes tend to be continuous, which provides a basis for separating ground and non-ground patches in its NGNEM. We used the maximum filtering tool with a 5 × 5 sliding window to analyze the NGNEM. Then, we calculated the binary raster R_{HC} as

$$R_{HC} = \begin{cases} 1 & \text{if } (GNEM_{\max} - GNEM) > W \\ 0 & \text{if } (GNEM_{\max} - GNEM) \leq W \end{cases} \quad (2)$$

The edges of non-ground objects can be outlined in the R_{HC}. Thus, we used the patch labelling tool based on the optimized Union-Find algorithm (Wu et al., 2005) to label each patch shown in an R_{HC} with a unique series number (Fig. 3C). For a labeled patch P_L, perform a raster filling analysis. If P_L could be filled, we computed the maximum elevation of its edge (H_{edge}) and the average elevation of its filled area (H_{fill}). If H_{fill} ≥ H_{edge}, P_L is in non-ground regions. Otherwise, P_L lies in a steep or terraced area with continuous elevation variation.

We found the NGNEM may show the nodewise elevation of some ND_{CEs} from regions covered by rooftops or tree canopies. Filling binary holes to detect their coverage patches is impractical, as the edges of these misclassified non-ground objects may be incomplete. To address this issue, we applied a morphological closing analysis to the labeled patches in the NGNEM (van der Walt et al., 2014) to binary NGNEM (NGNEM_{BL}). The window size w_{close} for morphological closing analysis is

$$w_{close} = \sqrt{N_{close}}, \quad (3)$$

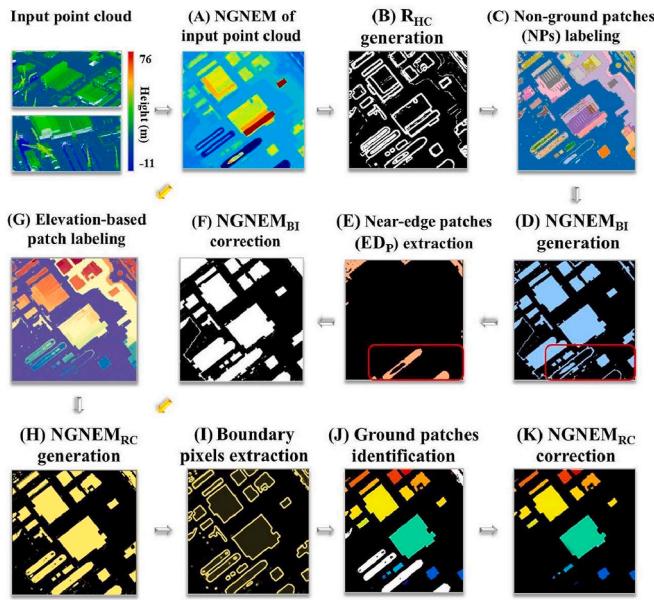


Fig. 3. Flowchart for separating ground and non-ground patches from the near-ground node elevation model (NGNEM). R_{HC} depicts the edges of non-ground objects. NGNEM_{RC} refers to the reclassified NGNEM, in which the colored regions represent non-ground patches. In subfigure (J), the white patches are identified as ground patches, which are removed in subfigure (K).

where N_{close} is the number of pixels within each closed patch. The closed patches covered by non-ground objects are marked with “1” in the NGNEM_{BI}, whereas the remaining pixels are marked with “0”.

To avoid some non-ground patches near the NGNEM edge missing in the NGNEM_{BI}, we reset all pixels in the NGNEM_{BI} to “0”, except for those in the first row, first column, last row, and last column of the NGNEM_{BI}; these are set to 1, 2, 3, and 4, respectively, to obtain the NGNEM_{edge}. Then, we analyzed ED_P using

$$ED_P = NGNEM_{BI} \otimes NGNEM_{edge}. \quad (4)$$

As shown in Fig. 3E, if a single patch in an ED_P has no more than two nonzero values, it is a patch near the edges of NGNEM. We marked these near-edge patches with “1” in the NGNEM_{BI}.

2.5. Patch labeling correction

In the NGNEM_{BI}, some ground patches on slopes are misclassified as non-ground type. To solve this error, we labeled patches in the NGNEM. The pixels in each NGNEM patch have the same elevation. In the related results, regions with the same elevation shown in the NGNEM form various patches. If pixels in a target patch within the labeled NGNEM are all “1” value in NGNEM_{BI}, it is assigned as a non-ground patch. Otherwise, the target patch in the labeled NGNEM is a ground patch. After reclassifying the labeled NGNEM, we obtained the NGNEM_{RC}.

Next, we applied morphological dilation analysis to patches with “1” value shown in the NGNEM_{RC} (Vanrell and Vitrià, 1997). The morphological dilation result of NGNEM_{RC} is then subtracted to itself to obtain the boundary pixel set of single patches (Fig. 3I). The boundary pixel set is used to extract the elevation variation range at each non-ground patch edge. If the maximum elevation at the edge of a non-ground patch exceeds the average elevation within its coverage area, it will be regarded as a ground patch on the slope or depression areas to correct the NGNEM_{RC} (Fig. 3 K).

2.6. Ground points extraction

According to the row and column index of pixels in the NGNEM_{RC}

and its spatial resolution W , the NGNEM_{RC} can be converted back into discrete nodes according to the center point of each pixel. The type of each converted node is determined according to pixel labeled in the NGNEM_{RC}: the TPN (TP_G) nodes with values of “1” (“0”) are within the non-ground (ground) patches. TP_G nodes provide the localization clues of ground points for the input point cloud. The elevation of each TP_G is determined by the node sharing the same (x, y) index in the ND_{NG}.

To improve the ground filtering accuracy of DMF, we used ND_{CES} to identify missing (x, y) index in the TP_G set as added nodes. The elevation of each added node is assigned to the average elevation of its 24 nearest TP_G nodes. In this way, a GF-support node layer can be constructed near the ground surface in the input point cloud. Then, we reduced the elevation of GF-support nodes by $v_1 \cdot W$ to avoid searching points from non-ground objects, such as vehicles parked on the road. According to the ablation test results, v_1 is set to 2.

In the next step, we built a KD-tree search engine (OKDT) for the input point cloud and another KD-tree search engine (TKDT) for the GF-support node layer to form a bidirectional KD-tree search engine (Fig. 4). For each point in the OKDT, we searched for its nearest GF-support node. Then, we counted the number of points in the input point cloud that share the same GF-support node as their nearest neighbor and calculated their average (M , $M \geq 1$). Next, we searched for M nearest-neighbor points from the input point cloud for each TP_G node (Fig. 4). For M points from the input point cloud sharing a nearest GF-support node, if the nearest GF-support node of a point is a TP_G, we recorded its index in the OKDT to obtain a list G_{index} . Finally, we extracted ground points from the input point cloud according to the G_{index} .

3. Experiments

3.1. Accuracy evaluation details

To assess the ground filtering accuracy of DMF and the other tested GFs, we counted the number of omitted and misclassified ground points from the GF results of 40 test point clouds selected in the OpenGF dataset (Qin et al., 2023b). Points in each selected point cloud is pre-labeled as either ground or non-ground type, providing the reference ground pointsets for evaluating the performance of tested GFs. Then, we calculated the overall accuracy (ACC), F-score, intersection over the union of class ground (IOU1), intersection over the union of class non-ground (IOU2), and Cohen’s Kappa coefficient (Kappa) to assess the ground filtering accuracy of test point clouds (Cao et al., 2024; Qin et al., 2023a; van der Walt et al., 2014) as follows:

$$ACC = \frac{TN + TP}{TP + FP + TN + FN}, \quad (5)$$

$$IOU1 = \frac{TP}{TP + FP + FN}, \quad (6)$$

$$IOU2 = \frac{TN}{TN + FP + FN}, \quad (7)$$

$$F-score = \frac{2TP}{2TP + FP + FN}, \quad (8)$$

$$Kappa = \frac{P_o - P_e}{1 - P_e}, \quad (9)$$

$$P_o = \frac{TP + TN}{TP + TN + FP + FN}, \quad (10)$$

$$P_e = \frac{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}{(TP + TN + FP + FN)^2}. \quad (11)$$

Each correctly classified ground (non-ground) point is a true positive (negative). The number of true positives, false positives, true negatives, and false negatives cases are called TP, FP, TN, and FN, respectively.

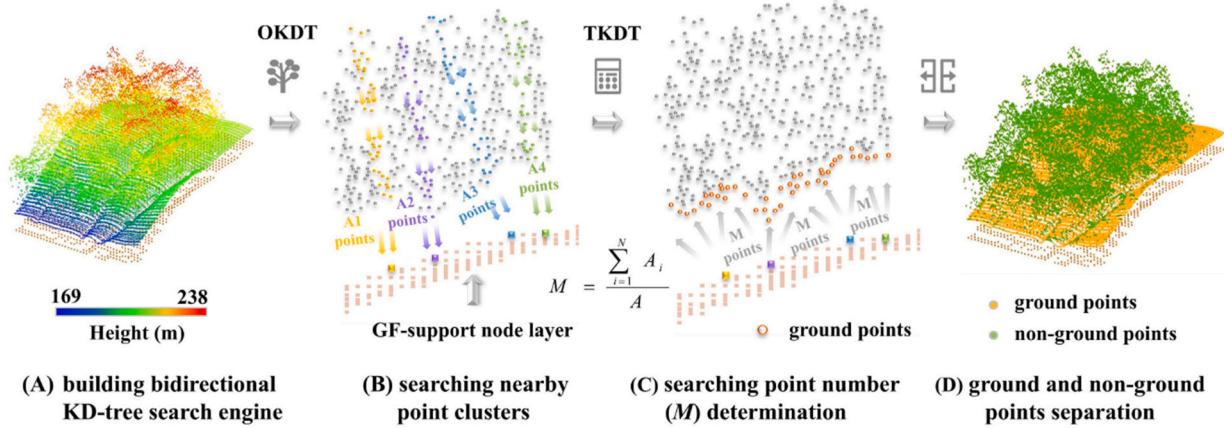


Fig. 4. Ground filtering processes run based on the GF-support node layer and a bidirectional KD-tree search engine. OKDT and TKDT mean the KD-tree search engine built upon the input point cloud and its GF-support nodes, respectively.

To evaluate the accuracy and robustness of DMF, we compared the ground filtering accuracy of its results with that of some open-source GFs. The tested rule-based GFs include CSF (Zhang et al., 2016), progressive morphological filter (PMF) (Keqi et al., 2003), skewness balancing filter (SBF) (Bartels and Wei, 2010), simple morphological filter (SMF) (Pingel et al., 2013), multiscale curvature classification method (MCC) (Evans and Hudak, 2007), improved progressive triangulation network densification method (TIN) (Zhao et al., 2016), and a slope-threshold-based GF tool (STF) embedded in Leica Cyclone 3DR software (<https://leica-geosystems.com/en-us/products/laser-scanners/software/leica-cyclone/leica-cyclone-3dr>). The MCC tool can be downloaded at <https://sourceforge.net/projects/mcclidar>. The TIN tool is embedded into the LiDAR360 software (<https://www.lidar360.com/>). The CSF tool can be tried using the EasyPoint software (<https://easy-point.xyz/download/>). The other rule-based GFs are implemented using the respective tools provided in the PDAL software (<https://pdal.io/en/2.7.2/stages/filters.html>) within the Python 3.8.1 environment.

We also selected four DL models, including PointNet++ (PNET2) (Qi et al., 2017), RSconv (Liu et al., 2019), KPconv (Thomas et al., 2019), and DGCNN (Wang et al., 2019), to test their performance in ground filtering. A set of 103 point clouds is selected from the OpenGF dataset (Qin et al., 2023b) to train these DL-based GFs, excluding the test datasets. These training point clouds were not used for testing selected GFs but to validate ground filtering accuracy of the proposed DMF. We used Pytorch v1.12.1 (Paszke et al., 2019), torch-points3d v1.3.0 (Paszke et al., 2019), and open3d-ml v0.18.0 (Chaton et al., 2020) packages to implement all tested DL models. The related experiments were performed on two GeForce RTX 3090 GPUs. Large point cloud segmentation models, such as SAM3D (Zhou et al., 2018) and SAM-Pro3D (Zhang et al., 2023), were excluded from the GFs comparison as they rely on pointwise RGB attributes and prompts to perform semantic segmentation of 3D objects.

High-density ground points lead to data redundancy in topographic representation, but they do not significantly impact the depiction of topographic features. Therefore, apart from ground filtering accuracy evaluation, we evaluated the accuracy of DTM generation using ground points extracted by GFs. Each DTM's spatial resolution is set to 2 m, with the pixel-level elevation calculated as the average elevation of GF-extracted points within each pixel. Similarly, each reference DTM (DTM_{ref}) is generated at the 2-m resolution using ground points labeled in the related point cloud.

The evaluation metrics of each GF-generated DTM are used to assess its generation completeness and accuracy. The generation completeness of DTM is evaluated using two metrics: the area ratio of misidentified non-ground regions (R_M) in the GF-generated DTM and that of omission ground coverage regions (R_O). Missing ground elevation information in

some non-ground regions, as well as plot altitude variation, may affect the DTM generation accuracy evaluation. Therefore, we used non-null elevation information within the intersection region of each DTM and its corresponding DTM_{ref} to evaluate the DTM generation accuracy, including the RMSE between DTM and DTM_{ref} (H_{RMSE}) and their average elevation difference (H_{DIF}): Table 1

$$H_{RMSE} = \sqrt{\frac{\sum_{i=1}^N (DTM - DTM_{ref})^2}{N}}, \quad (12)$$

$$H_{DIF} = \frac{\sum_{i=1}^N (DTM - DTM_{ref})}{N}, \quad (13)$$

where N represents the number of non-null pixels in the intersection region of DTM and DTM_{ref} (Qin et al., 2023a). We also calculated the average area ratio of pixels in the DTM with non-zero H_{DIF} s, which are categorized at four levels shown in Table 2.

Considering that the tested rule-based and DL-based GFs ran in different environments, we did not compare their efficiency. For the proposed DMF, we evaluated its running efficiency on the training dataset of DL-based GFs, measured by the number of points processed per second.

3.2. Benchmark dataset

To evaluate the accuracy and robustness of tested GFs, we randomly selected a dataset consisting of 40 point clouds from the OpenGF dataset (Fig. 5), and its detailed information is provided at <https://github.com/Nathan-UW/OpenGF> (Qin et al., 2023b). These point clouds are scanned from four types of scenarios: metropolis (ME), small city (SM), village (VI), and mountain (MO). Each point cloud in the test dataset has a spatial size of 500 m × 500 m, and no overlapping areas exist between adjacent blocks. The horizontal geodetic datum of selected ALS datasets is the New Zealand Geodetic Datum 2000, and the vertical geodetic datum is the New Zealand Vertical Datum 2016, adopted to ensure consistent orthometric heights. The pointwise semantic labels (ground or non-ground) for each point cloud were labeled in the original ALS datasets and validated by the OpenGF authors and by us during the test dataset selection, providing reference ground points for assessing the performance of GFs. The test dataset includes diverse types of noise to evaluate the robustness of GFs, such as high-value and low-value noise points, surface discontinuities, cliffs and ridges, artificial objects with complex structures (e.g., sunken buildings, towers, and power lines), vegetation-covered areas, and bridges. Table 1 records the point density

Table 1

The scene information of test point clouds. In the third column, a red upward arrow indicates the presence of high-value noise points, and a blue downward arrow represents the presence of low-value noise points.

Plot name	Point density (pts/m ²)	With noise	Bridge	Cliff or ridge	Sunken building	Shrub	Forest	Object complexity	Car or boat
MO (a)	16.23						●		
MO (b)	19.37						●		
MO (c)	13.14					●			
MO (d)	9.93	↑		●		●	●		
MO (e)	9.71			●		●	●		
MO (f)	11.15	↑				●			
MO (g)	12.89			●		●	●		
MO (h)	10.83	↑		●		●			
MO (i)	6.48	↑		●		●			
MO (j)	6.31	↑		●		●			
SM (a)	0.80					●			
SM (b)	0.96					●			
SM (c)	1.07			●		●			●
SM (d)	15.13					●			
SM (e)	18.12					●			
SM (f)	2.92		●		●	●		●	
SM (g)	3.02					●		●	
SM (h)	21.67	↑↓		●		●		●	●
SM (i)	22.36	↑↓				●	●	●	●
SM (j)	21.67	↑↓		●		●	●	●	●
ME (a)	16.91		●		●				●
ME (b)	17.18		●		●			●	●
ME (c)	16.48		●		●			●	●
ME (d)	15.86				●			●	●
ME (e)	16.45		●		●	●		●	●
ME (f)	18.08		●			●	●	●	●
ME (g)	16.01			●				●	●
ME (h)	23.94			●				●	●
ME (i)	12.76		●					●	●
ME (j)	15.07							●	●
VI (a)	0.58					●			●
VI (b)	0.95					●			●
VI (c)	0.91					●	●		
VI (d)	18.46		●			●	●		●
VI (e)	24.16						●		●
VI (f)	18.86					●	●		●
VI (g)	17.60					●	●		●
VI (h)	13.61					●	●		●
VI (i)	16.88					●	●		●
VI (j)	0.74					●	●		●

Table 2

The average error between the reference DTM of each test point cloud and the DTM generated from their GF results. The fourth and fifth columns show the average area ratio of omission ground regions (R_O) and misidentified non-ground patches (R_M) shown in DTM. The last four columns list the area ratio of pixels in the DTM with non-zero H_{DIFS} , categorized at four levels.

GFs	$\overline{H_{RMSE}}$ (m)	$\overline{H_{DIF}}$ (m)	$\overline{R_O}$ (%)	$\overline{R_M}$ (%)	<1 m (%)	[1m, 2 m](%)	[2m, 3m](%)	≥ 3 m (%)
DMF	0.55	0.13	2.18	2.15	97.38	2.17	0.32	0.13
CSF	0.62	0.22	2.88	2.28	94.01	4.14	1.31	0.54
MCC	0.79	0.27	5.05	4.08	96.41	2.66	0.56	0.37
PMF	1.95	0.98	16.37	1.06	94.33	4.06	0.63	0.99
SMF	0.59	0.21	3.10	3.93	95.22	3.41	0.85	0.53
TIN	0.62	0.20	2.96	2.61	98.78	0.88	0.18	0.15
SBF	5.97	4.91	11.45	4.95	78.84	7.63	2.41	11.12
STF	0.95	0.67	5.20	1.97	85.87	3.28	2.35	8.50
PNET2	7.36	4.59	5.39	12.99	77.81	2.50	1.32	18.37
KPconv	5.96	3.50	5.44	11.42	77.52	9.65	3.66	9.17
RSconv	2.00	0.59	10.26	7.18	90.96	3.19	1.11	4.74
DGCNN	0.99	0.41	6.18	6.82	86.90	8.26	3.14	1.70

and scene information of test point clouds.

4. Results

4.1. Ablation test results

We conducted an ablation test to determine v_0 and v_1 in DMF based on the average ground filtering accuracy of each point cloud used to train the DL-based GFs. The test range for two parameters is [1, 5], with a step size of 0.5. For the ground filtering accuracy evaluation, we focused more on IOU1 and F-score for the test combinations of v_0 and v_1 . We tested 81 combinations to compare the average IOU1, F-score, and IOU2 of 103 GF results for different parameter settings (Fig. 6). Based on a comprehensive evaluation, we set both parameters to 2.

According to Fig. 6, v_0 and v_1 jointly affect the ground filtering accuracy, with IOU1 being easily affected by v_0 and IOU2 being easily affected by v_1 . Increasing v_0 enlarges the voxel size, thereby improving the efficiency of DWF. However, enlarging v_0 blurs the structure details of an input point cloud, affecting the NGNEM generation. Besides, enlarging v_0 bothers the edge detection of non-ground objects shown in NGNEM, making it impossible to accurately identify the coverage regions of non-ground objects before marking semantic labels on the GF-support node layer. Setting overlarge v_1 may lead to incomplete extraction of ground points from slopes, while a v_1 that is too small may fail to remove point clouds from the surface of tiny non-ground objects, such as parked vehicles and roadside benches.

4.2. Ground point filtering accuracy

Due to the volume imbalance of ground and non-ground points within each test point cloud, we prioritized the IOU1, F-score, and IOU2 to evaluate the ground filtering accuracy. The results show that using DMF produces a higher average IOU1 (0.80), F-score (0.88), and IOU2 (0.86). The ground filtering accuracy of DMF for flat terrain is comparable to that of the CSF and SMF (Figs. 7 and 8). The average accuracy of these three GFs surpasses that of other tested GFs, demonstrating their resilience to variations in point density and abrupt height changes caused by non-ground objects, such as buildings and riverbanks. DMF can filter rooftop points out more accurately, as it relies on non-ground patches identified from NGNEM to identify their coverage area rather than distribution patterns extraction from local point clusters.

In mountainous scenarios, DMF shows superior ability and robustness in extracting ground points from steep slopes and ridges compared to the results of SBF, PMF, and STF (Figs. 7 and 8). For SMF, CSF, and MCC, using default parameters often leads to ground points loss from slopes, with the average F-score and IOU1 less than 0.48 in MO scenes. The TIN is less affected by terrain heterogeneity near steep slopes and ridges, while it tends to omit redundant ground points with an average F-score of data in MO scenes equal to 0.67. The errors in the DMF results

occur near the edges of point cloud coverage regions. Besides, some shrub points on slopes are misidentified as ground points in the DMF results.

We applied DMF to the training set of the tested DL-based GFs and recorded their accuracy and running efficiency. DMF achieved accuracy comparable to that on the test set, with an average F-score, IOU1, and IOU2 of 0.87, 0.80, and 0.84, respectively (Fig. 9). Under testing hardware conditions, DMF can process 3,6700 points per second on average.

The ground filtering accuracy of the tested DL-based GFs is not superior to that of the rule-based GFs, particularly in mountainous scenarios. The DL-based GFs demonstrate greater robustness in handling high-density point clouds, with average IOU1 and IOU2 values exceeding 0.74 for point clouds scanned in ME scenes. However, they struggle to accurately classify ground and non-ground points in regions such as slopes, areas with vegetation coverage, and rooftop surfaces, with an average F-score below 0.75 and an average IOU1 below 0.70. Feature extraction units of target point clouds built upon the k -nearest neighbor searching method are less effective for DL-based GFs than the local random sampling mechanism used in the PNET2 model. DL models are better suited for scenes with moderate and relatively uniform point density, as point density variations may bother local point clusters extraction when using the k -nearest neighbor searching method. Due to the difficulty in interpreting the DL-based ground filtering processes, we briefly analyzed the error sources caused by their architecture in the discussion section.

4.3. DTM generation accuracy

Ground points extracted by some GFs are not complete but correct, resulting in their IOU1 being lower than IOU2. For example, the average F-score and IOU1 of the TIN results are lower than 0.57, yet the extracted ground points are still suitable for generating high-quality DTM. The average H_{DIF} and H_{RMSE} of DTM generated using DMF, TIN, SMF, and CSF results are lower than 0.3 m and 0.7 m (Table 2). Besides, we used R_M and R_O to assess the area ratio of misidentified non-ground and missing ground patches in the GF-generated DTM. These patches are mainly distributed on steep slopes and in regions covered by rooftops. Among the tested rule-based GFs, the DTM derived from the DMF, TIN, and SMF results, with an H_{DIF} of less than 1 m, exceeds 95 %, and with an average R_O and R_M of less than 4 %. Thus, these GFs are appropriate for extracting ground points to generate DTM in mountainous scenes.

The average R_O of DTM generated using ground points extracted by PMF, SBF, MCC, and STF are over 5 %, due to data omission in the corresponding DTM. Visual inspection reveals that their GF results have large omission parts near slopes or areas with abrupt height changes, such as building edges and cliffs (Fig. 10). Although the ground points extracted by SMF have fewer omission parts, misidentified non-ground points from rooftops and tree canopy in their GF results increase the R_M . Using DMF achieved lower average R_O and R_M compared to the

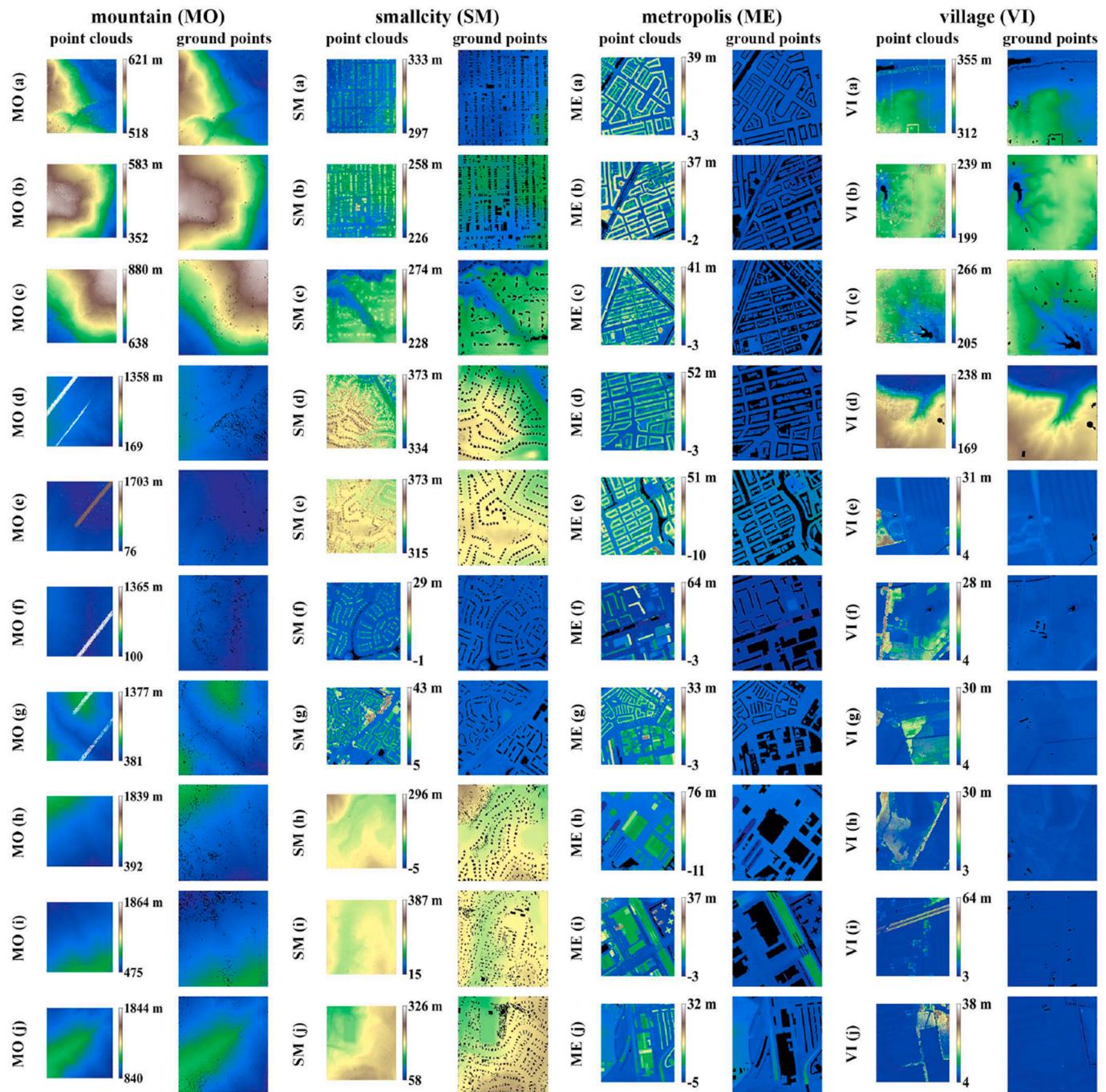


Fig. 5. The test dataset for selected GFs. In each pair of insets, the left one shows the elevation information of a test point cloud, and the right one presents the elevation variation of ground points in the same point cloud.

other GFs, with lower average H_{DIF} and H_{RMSE} for its generated DTM, demonstrating its applicability to various scenarios.

Using the DTM generated from the PNET2 and KPconv results is challenged to depict topographical features in test scenarios entirely and accurately (Table 2). The related average H_{RMSE} exceeds 5 m, and the average R_M is over 11 %, owing to the misidentified points from forest canopy or relatively flat rooftops in ground filtering results. Besides, we found some pixels with high H_{DIF} , primarily due to high-value outliers in point clouds that were not effectively removed in their GF results. The DTM generated from ground points extracted by DGCNN shows lower average H_{DIF} and H_{RMSE} when compared with the other test DL-based GFs. However, missing ground points in the DGCNN results limit its applicability for DTM generation. Additionally, the irregular

distribution of misidentified non-ground points, along with the omission of points on slopes, further restricts the use of DL-based GFs in DTM generation.

5. Discussion

5.1. Methods comparison

We evaluated the ground filtering accuracy and robustness of the selected GFs and DMF using datasets scanned from various scenarios. The factors affecting performance of the selected GFs are empirical threshold settings (e.g., slope, curvature, or height difference), inappropriate feature extraction unit settings of point clouds, and limited

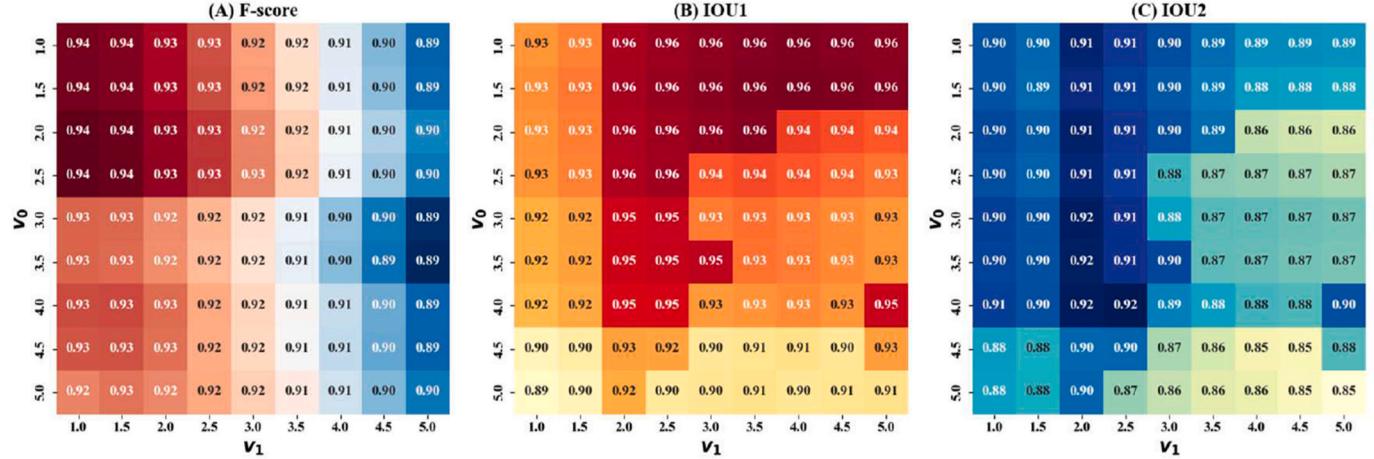


Fig. 6. The average accuracy comparison when setting different v_0 and v_1 in DMF.

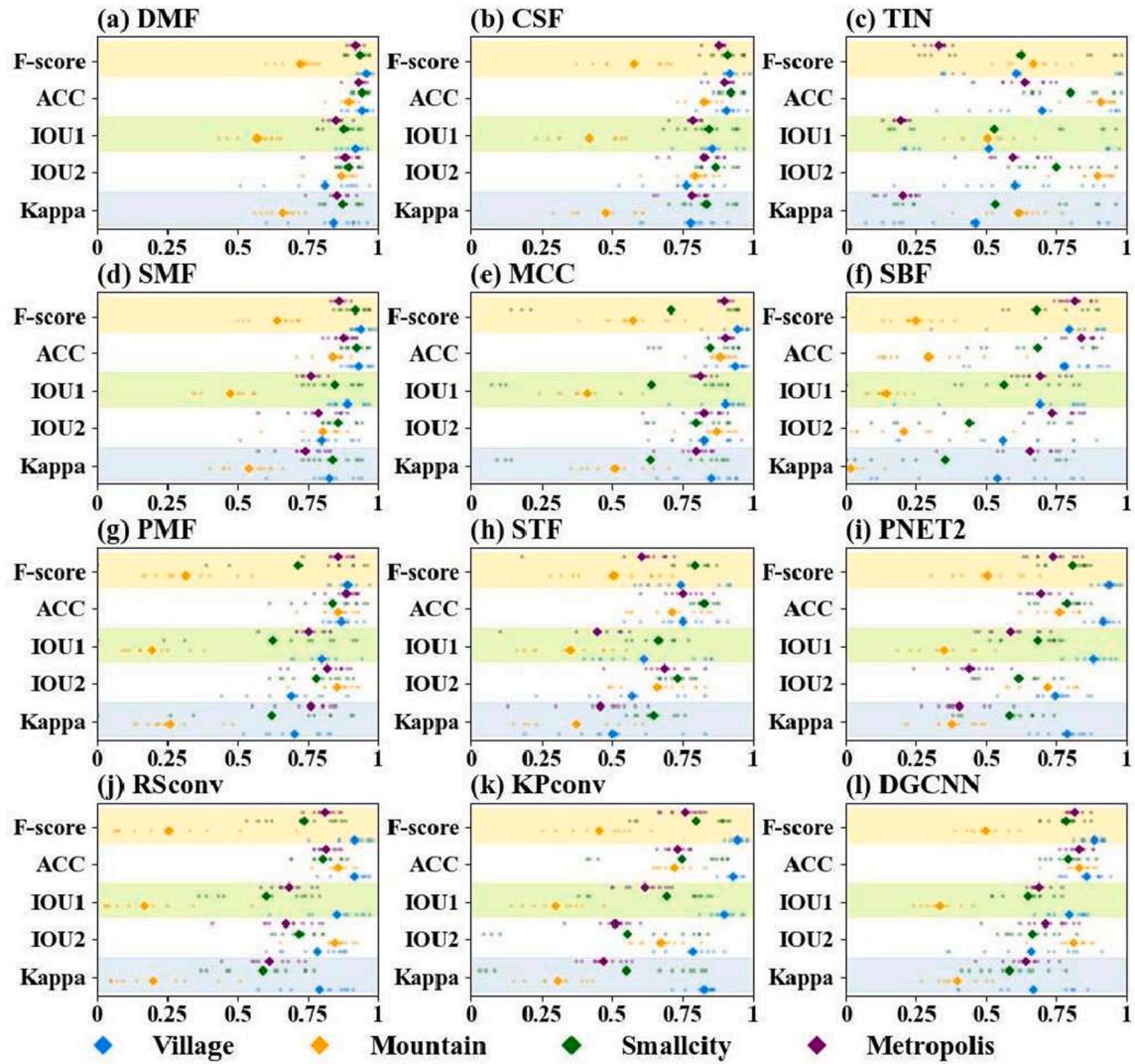


Fig. 7. The average accuracy of GFs for test point clouds. The evaluation metrics for the GF results include overall ground filtering accuracy (ACC), IOU1, IOU2, F-score, and Cohen's Kappa coefficient (Kappa), all with a range of [0, 1]. The colored points indicate the ground filtering accuracy for each test point cloud, and diamond dots representing the average evaluation metrics of test point clouds in the same scenario.

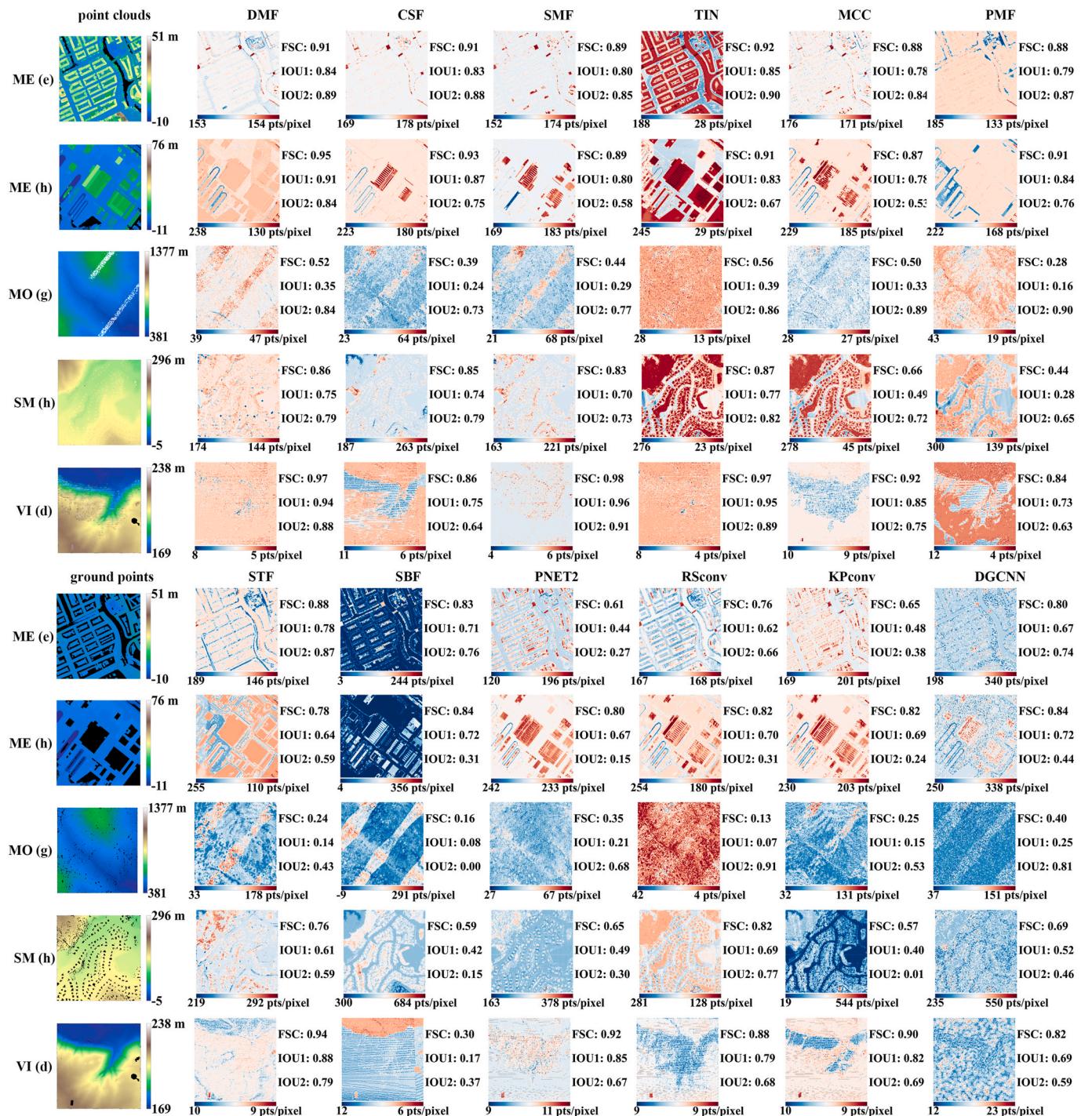


Fig. 8. The ground filtering results and accuracy of some test point clouds. The blue-red gradient color bar depicts the point density of omission (in blue) and misclassified ground points in the results of GFs. The FSC means the F-score.

perceptual field of encoder layers embedded in DL models. In some ME and SM scenarios, SMF, MCC, and PMF achieve ground filtering accuracy similar to that of DMF. When using SMF, the fixed window size for local point cluster extraction affects its accuracy. For MCC, selecting an appropriate range for local curvature assessment and threshold to separate ground and non-ground points is challenging, especially in areas covered by non-ground objects with complex structures (Zhao et al., 2018). Uneven point densities also affect curvature assessment (Nurunnabi et al., 2021; Qin et al., 2023b). PMF, which increases the window size progressively to analyze local point cluster distribution, may overlook detailed terrain features if the increments are too large,

leading to ground point omissions on slopes (Keqi et al., 2003).

Using CSF can obtain accurate ground filtering results when faced with redundant ground points. The key difference between CSF and other tested GFs lies in its construction of a virtual particle layer above the elevation-inverted point cloud. By simulating the fall of particles under gravity, this virtual particle layer is then used to fit the resampled set of inverted ground points, enabling ground filtering. The ground filtering process of CSF does not rely on predefined morphological filters or threshold settings based on local elevation skewness, surface curvature, or slope. However, data omission in the CSF results occurs in regions with steep slopes when the simulated “cloth” is not fully aligned

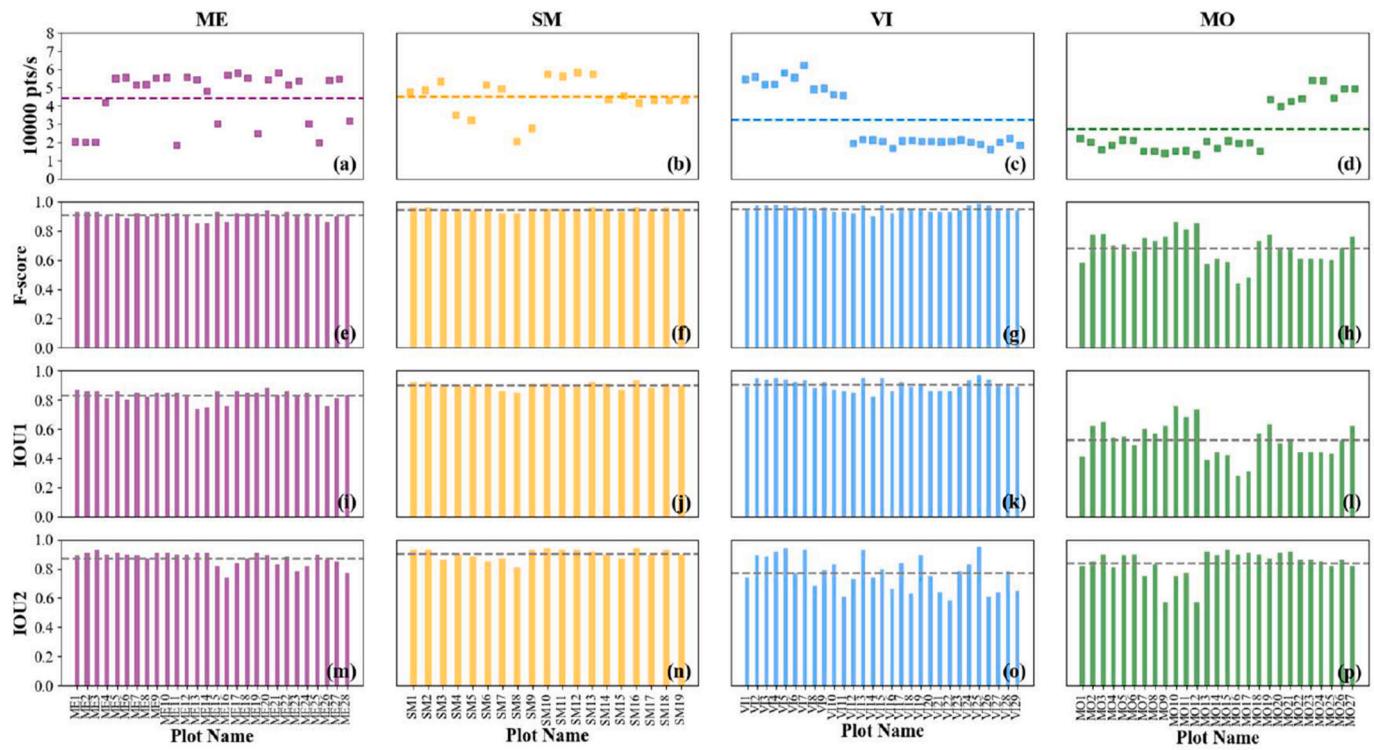


Fig. 9. Ground filtering accuracy and efficiency of DMF for the training datasets of the tested DL-based GFs. Colored blocks in the insets (a) to (d) indicate the number of processing points per second. Dashed lines denote the average value of five metrics.

with the top layer of inverted point clouds, leading to the height difference between slope surface and “cloth” exceeding the setting threshold during the ground filtering processes(Cai et al., 2024; Zhang et al., 2016).

TIN is more robust for mountainous scenes than DMF. We found that ground point omission remains in TIN-based results for high-density point clouds (Cai et al., 2019; Razak et al., 2011; Zhao et al., 2018). In addition, triangulated network construction is susceptible to noise points (Bulatov et al., 2021; Zhao et al., 2016). DMF uses the GF-support node layer, like near-ground triangulated network nodes, to ensure it shows similar advantages to TIN in processing point clouds scanned from mountainous scenarios.

SBF uses local elevation skewness to perform ground filtering automatically, assuming ground point distributions follow a normal pattern (Bartels and Wei, 2010). In areas with abrupt elevation changes, a non-normal distribution of ground points may lead to omissions in the SBF results. For example, SBF fails to extract points from the slope shown in the SM (h) (Fig. 8). Although STF can distinguish between ground and non-ground points when an appropriate slope threshold is applied, its robustness is limited by the window size used for local slope assessments.

Rule-based GF are interpretable, but setting an inappropriate feature extraction unit and empirical thresholds can affect their robustness and accuracy. Developing a DL-based GF is a novel and rapidly growing research field. Pre-trained DL models for 3D segmentation are typically trained on open-source datasets such as S3DIS and ScanNet, which are not specific to ground filtering. We cannot confirm the reliability and robustness of pre-trained DL models, as their training datasets may not include all real-life scenarios. Therefore, we trained them using the OpenGF dataset.

Since exploring optimal hyperparameters for DL-based GFs is not the focus of this study, we focused on how the feature extraction method affects their results. PNET2 leverages farthest point sampling and ball query to build hierarchical features, effectively capturing local and global geometric patterns, even with non-uniform point distribution

patterns (Qi et al., 2017). Rather than extracting features for discrete point clusters, DGCNN model employs k -nearest neighbors for dynamic graph construction and edge convolution to mitigate the effects of point density variations during 3D object segmentation (Wang et al., 2019). RSconv uses symmetric convolutional kernels to optimize neighborhood configurations and capture target cluster distributions regardless of orientation (Liu et al., 2019). KPconv arranges specialized kernel points to model fine-grained geometry of point clouds in detail to capture their local features (Thomas et al., 2019). However, none of the tested DL-based models achieved a balance between GF accuracy and completeness. More accurate DL-based GF results exhibit relatively higher average R_M and R_O . Thus, they struggle to fully depict the topographical features shown in the point clouds. Optimizing kernel point distributions in KPconv, fine-tuning neighborhood settings in DGCNN and RSconv, and implementing adequate hierarchical sampling in PNET2 may improve their accuracy.

Designing a DL-based GF is challenging due to uninterpretable feature extraction processes and empirical feature extraction unit settings (Cao et al., 2024). The distribution mode of point clusters near regions between ground and non-ground objects is always non-uniform. Local point cluster extracted by a DL-based GF for model training may fail to simultaneously obtain ground and non-ground points when using k -nearest neighbors or other similar methods. This limitation prevents DL models from properly learning distribution trends of point clusters to separate ground and non-ground points. Moreover, ground or non-ground point clusters may exhibit distribution patterns like those of mixed point clusters, leading to misclassification of points from flat rooftops or grasslands as ground points. Additionally, the distribution patterns of local point clusters from flat surfaces, such as large rooftops and bridges, are indistinguishable from those from ground surfaces. Optimizing DL model through error analysis is challenging because segmentation errors in their results lack clear patterns. Thus, the rule-based and interpretable GFs remain the preferred choice for users to obtain accurate and trustworthy ground filtering results.

The DMF algorithm identifies the coverage areas of non-ground

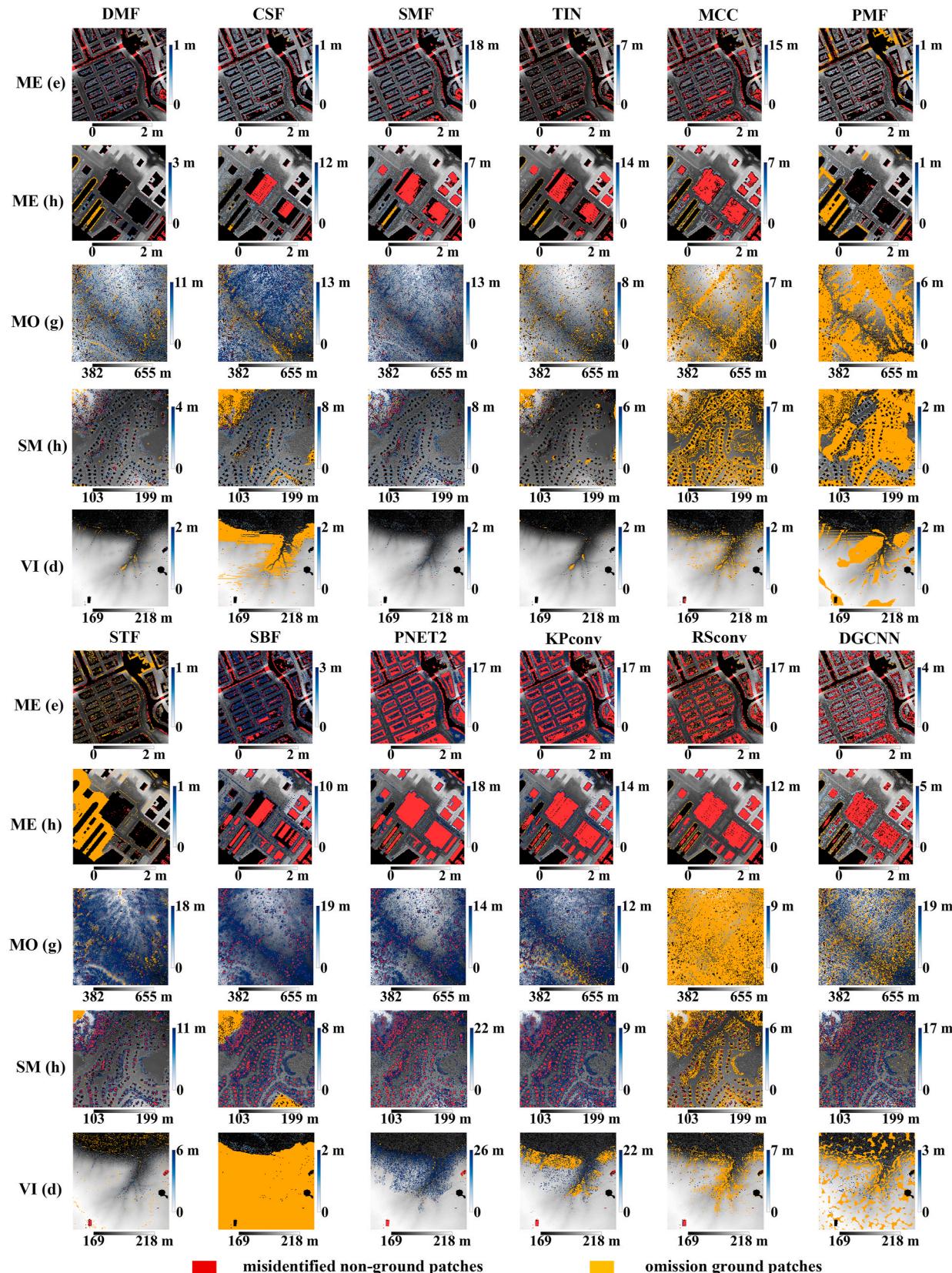


Fig. 10. Error distribution in the DTM generated from ground points extracted by the tested GFs. The grayscale background in each inset represents elevation variations in the DTM_{ref}. The gradient blue areas indicate the elevation differences between each DTM generated from GF results and its corresponding DTM_{ref}. Misidentified non-ground objects are highlighted with red patches, and orange patches represent areas covered by omission ground points.

objects by using a combined morphological filtering algorithm based on the NGNEM for separating ground and non-ground patches. Unlike other GFs that rely on fitting to local point cloud distribution patterns or triangulation mesh fluctuation trends to identify non-ground regions, our approach offers a raster-based method for labeling non-ground patches in the NGNEM, which is robust in areas with dense heterogeneous buildings and uneven point distributions.

While the DMF, like the TIN and CSF, automatically produces a GF-support node layer to assist in ground point filtering, it binarizes the digital elevation model formed by the near-ground voxel centroids and builds a bidirectional KD-tree search engine between the GF-support node layer and the related point cloud, which assists in ground point extraction. In contrast, other tested GFs require the setting of specific thresholds of elevation differences between ground and non-ground objects or local topographical features (e.g., slope, elevation skewness, or surface curvature). DMF does not require setting thresholds but uses a data-driven method to determine the number of search points for filtering ground points near each GF-support node, utilizing the KD-tree search engine to ensure robustness across various scenarios.

During the identification of non-ground patches from the NGNEM, the intermediate outputs of DMF can assign a unique label to each artificial object patch (Fig. 3 K), which can assist in instance segmentation of artificial objects from the input point cloud. This functionality is not currently available in the other tested GFs, which makes DMF valuable for supporting more advanced scene perception of point clouds, going beyond just ground point filtering.

DMF achieves an average F-score of 0.88. Moreover, the variation range of evaluation metrics for DMF results is generally narrower than that of the other tested GFs (Fig. 7). The average H_{RMSE} of the DTM generated from the DMF results is less than 0.6 m, with 97.38 % of pixels matching their reference ground elevation within a 1-m error. Our results confirm that DMF balances ground filtering accuracy, efficiency, and robustness across various scenarios. However, errors remain in the DMF results.

5.2. Error analysis

Misclassified points in the DMF results typically occur on slopes covered by dense shrubs and forests. Since DMF employs a bottom-up search method using a bidirectional KD-tree search engine and the GF-support node layer to identify ground points from the input point cloud, it may misclassify points from low shrubs or understory vegetation as ground points (Fig. 8). To improve ground filtering accuracy, DMF reduces the nodewise height of GF-support nodes to avoid the point clusters extraction from non-ground objects. Although this strategy may not eliminate all non-ground points, the extracted points align closely with the terrain undulation trend, minimizing their impact on the DTM generation and topographic feature analysis.

We set W as the elevation-difference threshold to support the extraction of non-ground patches within the NEDEM. It can be confirmed that the height difference between an artificial object and its nearby ground, as depicted in ALS point clouds, is greater than W , which provides an engine for extracting patches covered by artificial objects, such as rooftops and vehicles. However, point clusters from non-ground objects with no clear height difference from their surrounding ground surface, such as bridges and pedestrian overpasses, are easily misclassified as ground points, regardless of the tested GFs (Cao et al., 2024; Meng et al., 2010). These misclassified areas may be helpful in road connectivity analysis but negatively impact ground surface area calculations and land use mapping. Integrating bridge detection models can help optimize ground filtering results (Kim et al., 2020). Additionally, exploring the incorporation of other point cloud attributes, such as surface color and laser return intensity (Han et al., 2024; Rahman and Hoskere, 2024), into the GF process is also worth consideration.

The misclassified points from rooftops in the DMF results typically appear near the boundary of point clouds. Incomplete point clusters

from rooftops near data edges disrupt the height difference calculation between them and their surrounding ground surface, which causes some non-ground patches missing in the NGNEM_{BI}. DMF searches for M ground points near each GF-support node. If the number of ground points near a GF-support node exceeds M , some redundant ground points may be overlooked, reducing IOU1 but having a minimal impact on quality (Fig. 10). Similarly, ground point omission occurs in the TIN results, though it has little effect on the representation of topographic features shown in the ALS datasets.

During the morphological closing analysis for the NGNEM_{BI}, patches of some corridor-type buildings are filled in before labeling rooftop patches. Thus, atriums within building coverage area may be incorrectly identified as non-ground regions, as shown in the ME (h) of Fig. 8. We also found that DMF omits some ground points near building corners. Each pixel value in a NGNEM is calculated as the local average elevation of roof corners and their nearby ground surface. Thus, ground points near low rooftop edge are easily misclassified as non-ground points. Besides, the smoother local elevation variation of GF-support nodes, compared to the actual ground surface, leads to ground point omissions on ridges and cliffs. These missing ground points are farther from their nearest GF-support node than local non-ground points. To address this, we expanded the coverage area of GF-support nodes to reduce ground point omission on ridges and cliffs.

5.3. Applications

DMF converts a point cloud into a NGNEM to retrieve non-ground and ground patches. The data-driven voxel size (W) affects the edge extraction of non-ground patches in the NGNEM. When setting W , it is crucial to consider the height of non-ground objects and the distance measurement units (e.g., feet) of the target point cloud. Users should convert the distance units of the target point cloud to meters to reduce extraction errors of non-ground patches in the NGNEM.

DMF is less affected by noise points, as they do not influence the ND_{NG} generation. Therefore, point cloud denoising is not required when using DMF. Each labeled non-ground patch in the NGNEM provides a clue for extracting point clusters corresponding to individual artificial objects (e.g., vehicles, rooftops, boats, and overhead wires), which provides prompts to run large 3D segmentation models, such as SAM3D and SAMPro3D (Han et al., 2024; Xu et al., 2023). Once point clusters of artificial objects and ground are extracted from an input point cloud, the residual forest point clouds can be easily obtained to estimate multiscale vegetation coverage and canopy structure features (Liang et al., 2019; Montealegre et al., 2015). Thus, DMF provides a reliable engine for supporting rule-based semantic segmentation of ALS datasets.

6. Conclusions

The proposed DMF employs a data-driven approach to determine the voxel size before voxelizing an input point cloud, thereby extracting near-ground voxel centroids for generating the NGNEM. A composite morphological filtering method is then applied to identify ground patches in the NGNEM. This method can suppress misclassified non-ground patches with incomplete edge structures and errors caused by the loss of ground patches in slope and depression areas, labeling near-ground voxel centroids as ground or non-ground types to form a GF-support node layer. Finally, DMF uses a bidirectional KD-tree search engine built between the labeled GF-support nodes and the input point cloud to extract ground points, rather than relying on fixed elevation differences or topographical feature thresholds as filtering rules, which enhances its robustness across various test scenarios. Experimental results show that DMF achieves an average F-score exceeding 0.88. Furthermore, the intermediate results of DMF can support the instance segmentation of artificial objects, such as rooftops, depicted in the point clouds for urban modeling and land use mapping. However, some errors persist in DMF results, such as the loss of redundant ground points, the

loss of ground points near rooftop corners, and misclassification of points from non-ground objects with similar elevation to their nearby ground surface. Future work should explore combining DMF with non-spatial (color- or intensity-based) attributes of the input point cloud, or with specific identification models for artificial objects, to improve ground filtering accuracy and enhance the quality of topographical feature mapping.

CRediT authorship contribution statement

Bingxiao Wu: Writing – original draft, Software, Project administration, Methodology, Conceptualization. **Xingxing Zhou:** Validation, Project administration. **Junhong Zhao:** Validation, Project administration, Conceptualization. **Wuming Zhang:** Writing – review & editing. **Guang Zheng:** Writing – review & editing, Project administration.

Funding

This study was supported by the Key-Area Research and Development Program of Guangdong Province (grant Nos. 2023B0202090001), National Natural Science Foundation of China (grant Nos. 52472447 and grant Nos. 42171340), China Postdoctoral Science Foundation (grant Nos. 2022M723701), and the Scientific and Technological Innovation Strategic Program of Guangdong Academy of Agricultural Sciences (grant Nos. ZX202402).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We are very grateful to anonymous reviewers for their constructive suggestions. We sincerely thank data providers for sharing the OpenGF dataset. We would also like to express our gratitude to the developers of the tested GFs and the related software.

Data availability

Data will be made available upon request.

References

- An, Z., Sun, G., Liu, Y., et al., 2024. Rethinking few-shot 3D point cloud semantic segmentation. CoRR, 00592. <https://doi.org/10.48550/arXiv.2403.00592> abs/2403.
- Bai, Y., Zhang, B., Xu, N., et al., 2023. Vision-based navigation and guidance for agricultural autonomous vehicles and robots: a review. Comput. Electron. Agric. 205. <https://doi.org/10.1016/j.compag.2022.107584>.
- Bartels, M., Wei, H., 2010. Threshold-free object and ground point separation in LIDAR data. Pattern Recognit. Lett. 31, 1089–1099. <https://doi.org/10.1016/j.patrec.2010.03.007>.
- Bulatov, D., Stutz, D., Hacker, J., et al., 2021. Classification of airborne 3D point clouds regarding separation of vegetation in complex environments. Appl. Opt. 60, F6–F20. <https://doi.org/10.1364/AO.422973>.
- Cai, S., Pang, Y., Yu, S., et al., 2024. Filtering airborne LiDAR data based on multi-view window and multi-resolution hierarchical cloth simulation. Geo-Spatial Inf. Sci. 1–18. <https://doi.org/10.1080/10095020.2024.2354211>.
- Cai, S., Zhang, W., Liang, X., et al., 2019. Filtering airborne LiDAR data through complementary cloth simulation and progressive TIN densification filters. Remote Sens. 11. <https://doi.org/10.3390/rs11091037>.
- Cao, D., Wang, C., Du, M., et al., 2024. A multiscale filtering method for airborne LiDAR data using modified 3D alpha shape. Remote Sens. 16. <https://doi.org/10.3390/rs16081443>.
- Chaton, T., Chaulet, N., Horache, S., et al., 2020. Torch-Points3D: A Modular Multi-Task Framework for Reproducible Deep Learning on 3D Point Clouds, November 25–28, 2020[C]. IEEE, Japan, pp. 1–10. <https://doi.org/10.1109/3dv50981.2020.00029>.
- Chen, C., Chang, B., Li, Y., et al., 2021. Filtering airborne LiDAR point clouds based on a scale-irrelevant and terrain-adaptive approach. Measurement 171. <https://doi.org/10.1016/j.measurement.2020.108756>.
- Chen, Q., Wang, H., Zhang, H., et al., 2016. A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. Remote Sens. 8. <https://doi.org/10.3390/rs8010071>.
- Cheng, L., Hao, R., Cheng, Z., et al., 2023. Modeling the global relationship via the point cloud transformer for the terrain filtering of airborne LiDAR data. Remote Sens. 15. <https://doi.org/10.3390/rs15235434>.
- Dai, H., Hu, X., Zhang, J., et al., 2024. Large-Scale ALS point cloud segmentation via projection-based context embedding. IEEE Trans. Geosci. Rem. Sens. 62, 1–16. <https://doi.org/10.1109/tgrs.2024.3392267>.
- Du, L., Pang, Y., Wang, Q., et al., 2023. A LiDAR biomass index-based approach for tree- and plot-level biomass mapping over forest farms using 3D point clouds. Rem. Sens. Environ. 290. <https://doi.org/10.1016/j.rse.2023.113543>.
- Eitel, J.U.H., Höfle, B., Vierling, L.A., et al., 2016. Beyond 3-D: the new spectrum of lidar applications for earth and ecological sciences. Rem. Sens. Environ. 186, 372–392. <https://doi.org/10.1016/j.rse.2016.08.018>.
- Evans, J.S., Hudak, A.T., 2007. A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. IEEE Trans. Geosci. Rem. Sens. 45, 1029–1038. <https://doi.org/10.1109/tgrs.2006.890412>.
- Gharineiat, Z., Tarshu Kurdi, F., Campbell, G., 2022. Review of automatic processing of topography and surface feature identification LiDAR data using machine learning techniques. Remote Sens. 14. <https://doi.org/10.3390/rs14194685>.
- Guo, M.-H., Cai, J.-X., Liu, Z.-N., et al., 2021. PCT: point cloud transformer. Computational Visual Media 7, 187–199. <https://doi.org/10.1007/s41095-021-0229-5>.
- Han, X.-F., Jin, Y.-F., Cheng, H.-X., et al., 2023. Dual transformer for point cloud analysis. IEEE Trans. Multimed. 25, 5638–5648. <https://doi.org/10.1109/TMM.2022.3198318>.
- Han, X., Liu, C., Zhou, Y., et al., 2024. WHU-Urban3D: an urban scene LiDAR point cloud dataset for semantic instance segmentation. ISPRS J. Photogrammetry Remote Sens. 209, 500–513. <https://doi.org/10.1016/j.isprsjprs.2024.02.007>.
- Hu, X., Yuan, Y., 2016. Deep-learning-based classification for DTM extraction from ALS point cloud. Remote Sens. 8. <https://doi.org/10.3390/rs8090730>.
- Jin, S., Su, Y., Zhao, X., et al., 2020. A point-based fully convolutional neural network for airborne LiDAR ground point filtering in forested environments. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 13, 3958–3974. <https://doi.org/10.1109/jstars.2020.3008477>.
- Keqi, Z., Shu-Ching, C., Whitman, D., et al., 2003. A progressive morphological filter for removing nonground measurements from airborne LiDAR data. IEEE Trans. Geosci. Rem. Sens. 41, 872–882. <https://doi.org/10.1109/tgrs.2003.810682>.
- Kim, H., Yoon, J., Sim, S.H., 2020. Automated bridge component recognition from point clouds using deep learning. Struct. Control Health Monit. 27. <https://doi.org/10.1002/stc.2591>.
- Klápkště, P., Fogl, M., Barták, V., et al., 2020. Sensitivity analysis of parameters and contrasting performance of ground filtering algorithms with UAV photogrammetry-based and LiDAR point clouds. International Journal of Digital Earth 13, 1672–1694. <https://doi.org/10.1080/17538947.2020.1791267>.
- Li, B., Lu, H., Wang, H., et al., 2022. Terrain-net: a highly-efficient, parameter-free, and easy-to-use deep neural network for ground filtering of uav LiDAR data in forested environments. Remote Sens. 14. <https://doi.org/10.3390/rs14225798>.
- Li, F., Zhu, H., Luo, Z., et al., 2021. An adaptive surface interpolation filter using cloth simulation and relief amplitude for airborne laser scanning data. Remote Sens. 13. <https://doi.org/10.3390/rs13152938>.
- Li, Y., Wu, H., Xu, H., et al., 2013. A gradient-constrained morphological filtering algorithm for airborne LiDAR. Opt. Laser. Technol. 54, 288–296. <https://doi.org/10.1016/j.optlastec.2013.06.007>.
- Liang, X., Wang, Y., Pyörälä, J., et al., 2019. Forest in situ observations using unmanned aerial vehicle as an alternative of terrestrial measurements. Forest Ecosystems 6. <https://doi.org/10.1186/s40663-019-0173-3>.
- Liu, Y., Fan, B., Xiang, S., et al., 2019. Relation-Shape Convolutional Neural Network for Point Cloud Analysis, June 16–20[C]. IEEE, Long Beach, CA, USA, pp. 8895–8904.
- Maguya, A.S., Junttila, V., Kauranne, T., 2013. Adaptive algorithm for large scale dtm interpolation from lidar data for forestry applications in steep forested terrain. ISPRS J. Photogrammetry Remote Sens. 85, 74–83. <https://doi.org/10.1016/j.isprsjprs.2013.08.005>.
- Meng, X., Currit, N., Zhao, K., 2010. Ground filtering algorithms for airborne LiDAR data: a review of critical issues. Remote Sens. 2, 833–860. <https://doi.org/10.3390/rs2030833>.
- Mongus, D., Žalík, B., 2012. Parameter-free ground filtering of LiDAR data for automatic DTM generation. ISPRS J. Photogrammetry Remote Sens. 67, 1–12. <https://doi.org/10.1016/j.isprsjprs.2011.10.002>.
- Montealegre, A.L., Lamelas, M.T., de la Riva, J., 2015. A comparison of open-source LiDAR filtering algorithms in a mediterranean forest environment. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 8, 4072–4085. <https://doi.org/10.1109/jstars.2015.2436974>.
- Nie, S., Wang, C., Dong, P., et al., 2017. A revised progressive TIN densification for filtering airborne LiDAR data. Measurement 104, 70–77. <https://doi.org/10.1016/j.measurement.2017.03.007>.
- Nurunnabi, A., Teferle, F.N., Li, J., et al., 2021. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. Int. Arch. Photogram. Rem. Sens. Spatial Inf. Sci. XLIII-B1–2021, 31–38. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2021-31-2021>.
- Orhan, O., 2021. Land suitability determination for citrus cultivation using a GIS-based multi-criteria analysis in Mersin, Turkey. Comput. Electron. Agric. 190. <https://doi.org/10.1016/j.compag.2021.106433>.
- Paszke, A., Gross, S., Massa, F., et al., 2019. PyTorch: an Imperative Style, High-Performance Deep Learning Library, pp. 8024–8035. Vancouver, BC, Canada.

- Pingel, T.J., Clarke, K.C., McBride, W.A., 2013. An improved simple morphological filter for the terrain classification of airborne LiDAR data. *ISPRS J. Photogrammetry Remote Sens.* 77, 21–30. <https://doi.org/10.1016/j.isprsjprs.2012.12.002>.
- Qi, C.R., Yi, L., Su, H., et al., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Long Beach, CA, USA, pp. 5099–5108. December 4-9 [C].
- Qin, L., Wu, W., Tian, Y., et al., 2017. LiDAR filtering of urban areas with region growing based on moving-window weighted iterative least-squares fitting. *IEEE Geoscience and Remote Sensing Letters* 14, 841–845. <https://doi.org/10.1109/lgrs.2017.2682854>.
- Qin, N., Tan, W., Guan, H., et al., 2023a. Towards intelligent ground filtering of large-scale topographic point clouds: a comprehensive survey. *Int. J. Appl. Earth Obs. Geoinf.* 125. <https://doi.org/10.1016/j.jag.2023.103566>.
- Qin, N., Tan, W., Ma, L., et al., 2023b. Deep learning for filtering the ground from ALS point clouds: a dataset, evaluations and issues. *ISPRS J. Photogrammetry Remote Sens.* 202, 246–261. <https://doi.org/10.1016/j.isprsjprs.2023.06.005>.
- Rahman, A.U., Hoskere, V., 2024. Instance segmentation of reinforced concrete bridges with synthetic point clouds. CoRR, 16381. <https://doi.org/10.48550/arXiv.2409.16381> abs/2409.
- Razak, K.A., Straatsma, M.W., van Westen, C.J., et al., 2011. Airborne laser scanning of forested landslides characterization: terrain model quality and visualization. *Geomorphology* 126, 186–200. <https://doi.org/10.1016/j.geomorph.2010.11.003>.
- Su, W., Sun, Z., Zhong, R., et al., 2015. A new hierarchical moving curve-fitting algorithm for filtering lidar data for automatic DTM generation. *Int. J. Rem. Sens.* 36, 3616–3635. <https://doi.org/10.1080/01431161.2015.1065356>.
- Thomas, H., Qi, C.R., Deschaud, J.-E., et al., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. IEEE, Seoul, Korea (South), pp. 6410–6419. <https://doi.org/10.1109/ICCV.2019.000651>, October 27 - November 2[C].
- van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., et al., 2014. scikit-image: image processing in Python. *PeerJ* 2. <https://doi.org/10.7717/peerj.453>.
- Vanrell, M., Vitrià, J., 1997. Optimal 3×3 decomposable disks for morphological transformations. *Image Vis Comput.* 15, 845–854. [https://doi.org/10.1016/S0262-8856\(97\)00026-7](https://doi.org/10.1016/S0262-8856(97)00026-7).
- Wang, Y., Sun, Y., Liu, Z., et al., 2019. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* 38 (146), 141–146:112. <https://doi.org/10.1145/3326362>.
- Wei-Lwun, L., Murphy, K.P., Little, J.J., et al., 2009. A hybrid conditional random field for estimating the underlying ground surface from airborne LiDAR data. *IEEE Trans. Geosci. Rem. Sens.* 47, 2913–2922. <https://doi.org/10.1109/tgrs.2009.2017738>.
- Wu, K., Otoo, E.J., Shoshani, A., 2005. Optimizing Connected Component Labeling Algorithms. <https://doi.org/10.1117/12.596105>, 29 April 2005[C]. San Diego, California, United States: SPIE.
- Wu, Y., Liu, J., Gong, M., et al., 2024. MPCT: multiscale point cloud transformer with a residual network. *IEEE Trans. Multimed.* 26, 3505–3516. <https://doi.org/10.1109/TMM.2023.3312855>.
- Xu, M., Yin, X., Qiu, L., et al., 2023. SAMPro3D: locating SAM prompts in 3D for zero-shot scene segmentation. CoRR, abs/2311.17707. <https://doi.org/10.48550/ARXIV.2311.17707>.
- Zhang, D., Liang, D., Yang, H., et al., 2023. SAM3D: zero-shot 3D object detection via segment anything model. CoRR, abs/2306.02245. <https://doi.org/10.48550/arXiv.2306.02245>.
- Zhang, W., Qi, J., Wan, P., et al., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* 8, 501. <https://doi.org/10.3390/rs8060501>.
- Zhao, L., Liu, G., Lu, Z., et al., 2024. A new framework for delineating farmland consolidation priority areas for promoting agricultural mechanization in hilly and mountainous areas. *Comput. Electron. Agric.* 218. <https://doi.org/10.1016/j.compag.2024.108681>.
- Zhao, X., Guo, Q., Su, Y., et al., 2016. Improved progressive TIN densification filtering algorithm for airborne LiDAR data in forested areas. *ISPRS J. Photogrammetry Remote Sens.* 117, 79–91. <https://doi.org/10.1016/j.isprsjprs.2016.03.016>.
- Zhao, X., Su, Y., Li, W., et al., 2018. A comparison of LiDAR filtering algorithms in vegetated mountain areas. *Can. J. Rem. Sens.* 44, 287–298. <https://doi.org/10.1080/07038992.2018.1481738>.
- Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: a modern library for 3D data processing. CoRR. <https://doi.org/10.48550/arXiv.1801.09847> abs/1801.09847.