2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

A Self-organized Maps Ground Extract Method based on Principal Component Analysis

Yu Yao, Yunhua Li, *Senior Member, IEEE,* and Tao Qin

Abstract— **The lightweight of point clouds is an essential issue for LiDAR in practical applications. Point clouds collected outdoors often have a large number of ground points, reducing the data processing speed and affecting the classification and identification of targets. The paper develops a ground extraction method based on principal component analysis (PCA) and self-organizing map (SOM). The sufficient information is selected by analyzing the original point cloud features to improve the statistical outlier removal filter to achieve the initial cleaning of the point cloud. The filtered point cloud is reduced dimension by PCA, and overcomes the feature classification difficulty while accelerating the subsequent point cloud processing. Furthermore, SOM achieves unsupervised learning for the practical point cloud, which performs efficient ground extraction at sparse and dense locations while not relying on the size of the dataset. Experiments on SemanticKitti show that the detection accuracy of the proposed method can reach 95%, and it also has the satisfactory real-time performance.**

Index Terms*—*point cloud, ground separation, filter, principal component analysis, self-organizing map

I. Introduction

In practical applications, LiDAR mainly works in scenes such as factories, urban roads, construction sites, fields, etc. There is a huge number of ground points in these scenes, which can reduce the data processing speed and affect the classification and recognition accuracy of targets. Accurate separation and identification of ground points from point cloud data can reduce the amount of computing, improve the algorithm's efficiency, and play an essential role in improving the accuracy of environmental sensing.

In addition to the vast amount of data, point cloud has three characteristics: disorder, spatiality, and invariance. Disorder means that the point cloud consists of a large number of point data, and these points are stored in the file without order; spatiality signifies that points contain spatial coordinate information, which constitutes a set of spatial relationships; invariance implies that the target does not change with the rotation and translation of the point cloud data. It is necessary to overcome the difficulties caused by the above characteristics while using them for environmental perception during extracting ground information from point clouds. Sithole et al. [1] set the optimal slope threshold by a priori knowledge of the experimental area based on the principle that the slope values between ground points are small, while the slope values between ground and non-ground points are large, and achieved simple separation of the ground. However, the terrain is usually complex and variable, and it is unreasonable to set a uniform slope threshold for the whole experimental area. Susaki [2] introduced a rough numerical ground model to address the above shortcomings and improved the separation accuracy of the algorithm in complex terrain by varying the slope threshold. Lin et al. [3] used the region growing method to partition the point cloud into different parts, then set the rules to select the ground seed points to establish the initial irregular triangular network, and obtained the final ground point cloud by iterative filtering. Axelsson [4] created a sparse ground seed point at a low-relatively set of points and refined the reference ground surface by progressive densification of the seed points. Chen [5] organized the point cloud through square grids, introduced preliminary information to construct the initial sparse triangular ground seeds, and extracted all ground points during the density interaction. Although the method of presetting ground seed points has improved performance compared to classification based on slope thresholds, it cannot effectively separate the low objects on the ground. Chen et al. [6] used multiscale morphology and Z-score to select ground seeds as correctly as possible and incorporated residual thresholds for topographic adaptation to various terrain features. Bayram et al. [7] proposed a ground extraction algorithm based on spectral features to achieve high-precision extraction of bridge ground by analyzing 3D point clouds through weighted spectral maps.

In particular, it should be noted that point cloud filtering has a significant impact on improving the accuracy of feature extraction, which is usually combined with ground extraction in most cases. Filtering is the representative step in the point cloud pre-processing, which needs to remove the noise from the 3D point cloud model while maintaining the geometric features of the model itself. Zou et al. [8] proposed a wavelet function bilateral filtering method, which obtained the distance between two points in the optimal neighborhood by particle swarm optimization method, then performed wavelet smoothing and preserved the edge eigenvalues through the Gaussian function. The technique had a denoising accuracy of 96.96% and could retain the complete feature points, but the filtering time was too long for practical applications. Ren et al. [9] established a multiscale noise removal overall filtering algorithm. The filter constructed a surface change factor based on the point's normal vector, which can determine whether the sampling points belong to the mutation region in the domain. The method removed the large-scale noise in the flat region by statistical filtering and deleted the small-scale noise in the mutation region by bilateral filtering. Li et al. [10] used an adaptive bilateral filter to scale the model, which effectively preserved the edge feature values, but the adaptive configuration of this method takes too long, and some points are lost when scaling the model. Elhoseny et al. [11] solved the problem of point clouds contaminated by different types of noise through the optimal BF model and MI classification denoising method. However, the method is computationally intensive, and the filtering process spends too much time. This paper aims to develop a ground separation

This work is supported by National Natural Science Foundation of China and Shanxi Coal-based Low Carbon Joint Fund under the grant No. U1910211. (*Corresponding author: Yunhua Li, e-mail:yhli@buaa.edu.cn.*)

Yu Yao, Yunhua Li and Tao Qin are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China.

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

method for real-field point clouds of dimensionality reduction based on principal component analysis and self-organizing mapping. In order to optimize the separation effect, point cloud information analysis and filtering algorithm improvement are used for point cloud pre-processing. The proposed method is tested on the urban road.

II. Point Cloud Captured by LiDAR

As shown in Fig. 1, the actuator of LiDAR is usually composed of three major parts: the laser transmitter, the scanning receiver and the photoreceptor [12]. The laser transmitter emits multiple laser beams. The scanning receiver receives the return laser after the laser encounters an obstacle and records the time and angle for positioning. The photoreceptor records the intensity of the returned laser.

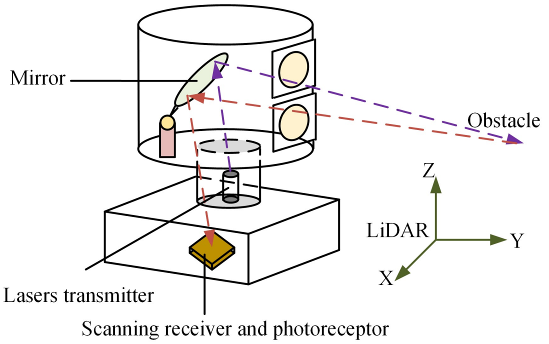


Fig. 1 The structure of LiDAR

A majority of LiDAR sensors use time-of-flight (ToF). The laser transmitter emits pulses and records the time and direction of emission; the pulses reflect part of the energy if encounter obstacles in the air. The receiver receives part of the reflected energy and records the time and power received. The distance is calculated as follows:

$$d = \frac{ct}{2} \tag{1}$$

where c represents the speed of light and t represents the round trip time. TABLE I. lists the resulting point cloud contains the following information for each point.

TABLE I. The Information for Each Point

Parameter (unit)	Implication
id_{p}	The serial number of the point
X, Y, Z(m)	The spatial coordinates of the point
A_{angle}(degree)	The clockwise horizontal azimuth of the point with LiDAR as the origin
A_{time}	The adjusted time of the point
d(m)	The distance from the point to LiDAR
I	The reflection intensity of the point
id_{L}	The laser beam of the point
T	The timestamp of the point
V(degree)	The pitch angle of the point with LiDAR as the origin

Due to the disorder, the id_{p} of the point cloud's point is just a representation form, which does not have practical significance and does not need in the perception. X, Y and Z represent the specific location information of points. A_{angle}, D and V can all be calculated by them according to the following equations:

$$D = \sqrt{X^2 + Y^2 + Z^2} \tag{2}$$

$$A_{\mathrm{angle}} = \arctan \frac{X}{Y} \tag{3}$$

$$V = \arccos \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} = \arctan \frac{Z}{\sqrt{X^2 + Y^2}} \tag{4}$$

Take Velodyne VLY-128E lidar for example, the hardware design of LiDAR fixes its beam distribution. Fig. 2 is the beam distribution of VLY-128E, the *laser ID* of each point can be queried according to A_{angle} and V.

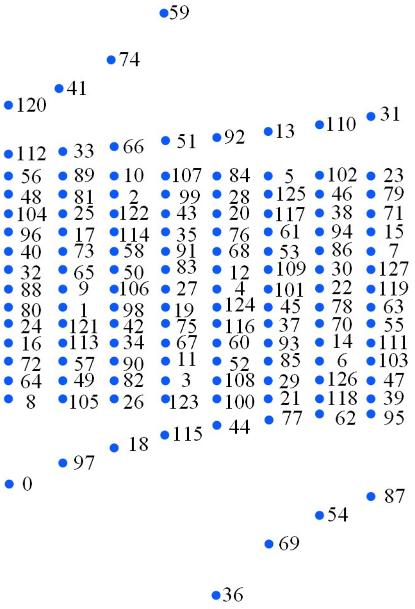


Fig. 2 The beam distribution of VLY-128E

I is a discrete integer value obtained from the comprehensive processing of the return optical power at the receiver. The value can be briefly expressed as follows：

$$I = h\left[K \cdot f(R) \cdot g(\theta)\right] \tag{5}$$

where h is the conversion function of the LiDAR signal, and K is the product of the target reflection coefficient and other system parameters. $f(R)$ is the distance factor to characterize the variation of the reflected intensity with distance, $g(\theta)$ is the angle factor represents the spatial distribution of the reflected intensity of the target.

The generation time of different points in each point cloud is different during the scanning process of LiDAR. T records the time of each point generation, and A_{time} can be used to correct the distortion generated by the scan time difference. Unlike most point cloud processing methods that only pay attention to spatial coordinates (X, Y, Z), the paper retains I for point cloud pre-processing and feature extraction to fully use the collected environmental features. Since this paper does not involve distortion correction of the point cloud, T and A_{time} are not processed.

III. Statistical Outlier Removal Filter that Adds Intensity

Due to the physical limitations of the sensor and the scene characteristics, LiDAR is extremely prone to forming noisy and discrete points during data acquisition, which contaminates the original point cloud.

Each point in the point cloud carries a certain amount of information, and the denser the points in a certain area, the more information is available. Statistical outlier removal filter calculates the average distance between a point and its *k* neighboring points and denotes it as *d*. Assuming that the set *D* composed of all *d* satisfies Gaussian distribution, the points whose *d* is outside the standard range of the sample are defined as outliers and should be removed from the point cloud.

Statistical outlier removal filter [13] merely takes advantage of *X*, *Y* and *Z* information, ignoring intensity information. The paper introduces intensity to calculate *d*, which further improves the filtering effect by considering the spatial distribution while combining the scene object surface information. The improved *d* and its threshold value d_{thresh} can be calculated as follows:

$$d = \sqrt{(X_i - X_{k_j})^2 + (Y_i - Y_{k_j})^2 + (Z_i - Z_{k_j})^2 + (I_i - I_{k_j})^2} \quad (6)$$

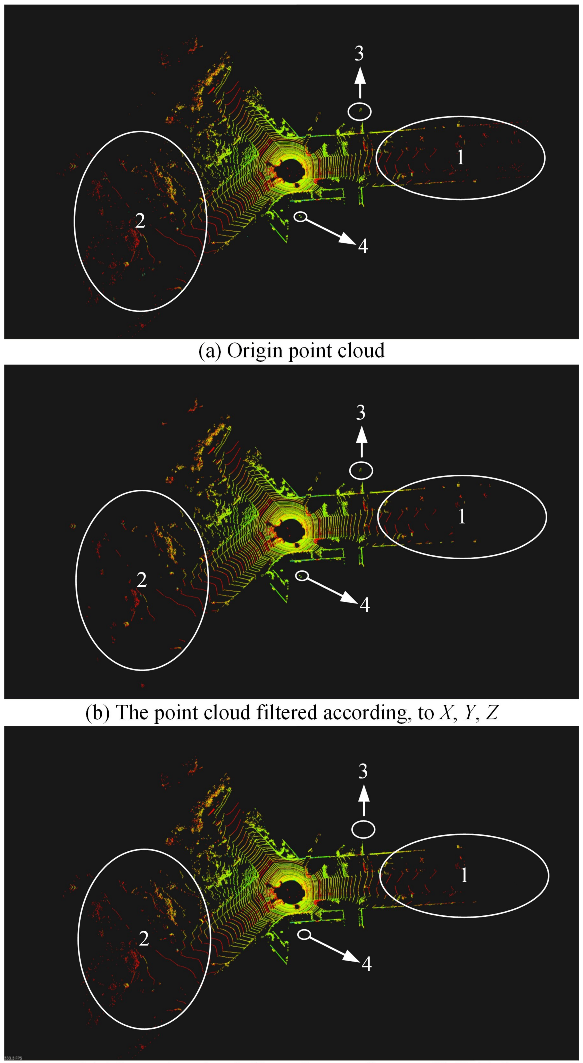
$$d_{thresh} = \mu + a * \sigma \quad (7)$$

where X_i, Y_i, Z_i, I_i and X_{k_j}, Y_{k_j}, Z_{k_j}, I_{k_j} respectively represent the coordinate and intensity of the *i*-th point and the *j*-th point in its *k* neighborhood, μ is the average distance, σ is the standard deviation of the distribution, and a is the proportionality constant, which can be adjusted during use.

Fig. 3 shows the effect of the two filters. From the white circle-3 and circle-4, it can be seen that the improved filter can effectively improve the discrete point removal at dense point clouds. The performance at sparse point clouds is slightly degraded, which is reflected in white circle-1 and circle-2.

IV. Dimension Reduction Based on Principal Component Analysis

The filtered point cloud removes unnecessary outlier points, which lays the foundation for efficient and accurate point cloud feature extraction. However, due to the large amount of data, the filtered point cloud still faces the problems of excessive computational pressure and poor real-time performance during processing. The paper uses 43552 point clouds of SemanticKitti [14], which the University of Bonn developed, to perform correlation analysis on the variables (*X*, *Y*, *Z*, *I*) screened in the previous section. The correlation coefficients between the variables are shown in Fig. 4, which shows that although the correlation between each variable is small, it is still not completely eliminated and provides overlapping information. Principal component analysis [15] is a multivariate statistical analysis method that transforms multiple correlated variables to a few linearly uncorrelated variables by rotating the axes of the original data onto orthogonal axes, which can be used to reduce the dimensionality of data and improve the efficiency of point cloud feature extraction.



(a) Origin point cloud

(b) The point cloud filtered according, to *X*, *Y*, *Z*

(c) The point cloud filtered according, to *X*, *Y*, *Z*, *I*

Fig. 3 The effects of different filters

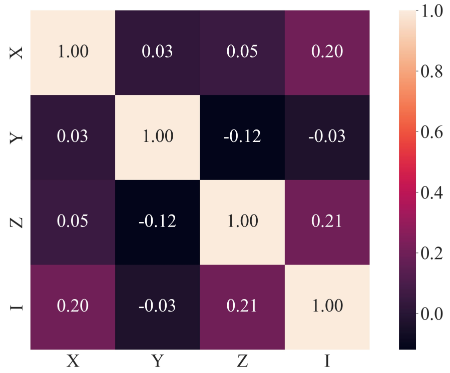


Fig. 4 The correlation coefficients between *X*, *Y*, *Z*, *I*

A. Mathematical Model of PCA

There are *n* samples, X_1, X_2, …, X_p are characteristic parameters of each sample without restriction on the unit. Each parameter may be correlated with others parameters. These parameters constitute the original data array as follows:

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}=(X_1, X_2, \cdots, X_p) \qquad (8)$$

Linearly combining the p column vectors X_1, X_2, …, X_p of X to obtain the integrated feature vectors as follows:

$$\begin{cases} F_1 = a_{11}X_1 + a_{21}X_2 + \cdots + a_{p1}X_p \\ F_2 = a_{12}X_1 + a_{22}X_2 + \cdots + a_{p2}X_p \\ \quad \cdots\cdots \\ F_p = a_{1p}X_1 + a_{2p}X_2 + \cdots + a_{pp}X_p \end{cases} \qquad (9)$$

where the coefficient $a_i^T=(a_{1i}, a_{2i}, \ldots, a_{pi})$ satisfies $a_i^T a_i=1(i=1, 2, \ldots, p)$, F_i is independent of each other ($\mathrm{Cov}(F_i, F_j)=0$, $(i\neq j)$), and $\mathrm{Var}(F_1)\geq\mathrm{Var}(F_2)\geq\ldots\geq\mathrm{Var}(F_p)$. Denote the covariance matrix of X as Σ. λ_1, λ_2, …, λ_p are the eigenvalues of Σ, and $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_p$. The standard orthogonal vectors corresponding to the eigenvalues are γ_1, γ_2, …, γ_p. The maximum value of $\mathrm{Var}(F_i)$ under the constraints of the above conditions can be calculated as follows:

$$\begin{aligned} \max(Var(F_i)) &= Var(F_1)=\max(Var(a_i^T X)) \\ &= \max\left(\frac{\gamma^T \sum \gamma}{\gamma^T \gamma}\right) = \lambda_1 = \frac{\gamma_1^T \sum \gamma_1}{\gamma_1^T \gamma_1} \end{aligned} \qquad (10)$$

Obviously, $a_i=\gamma_1$. Similarly, the integrated feature vectors as formulated in (11):

$$\begin{cases} F_1 = \gamma_1 X \\ F_2 = \gamma_2 X \\ \quad \cdots\cdots \\ F_p = \gamma_p X \end{cases} \qquad (11)$$

The variance contribution of F_i as formulated in (12):

$$C_i = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots \lambda_p} \quad (i = 1, 2, \cdots p) \qquad (12)$$

B. Dimensionality Reduction of Point Cloud

The paper analyzes the *XYZI* data of the point cloud with PCA. The statistics of the cumulative contribution of each principal component factor are shown in Fig. 5, which illustrates that retaining 2-dimensional data can effectively represent the point cloud.

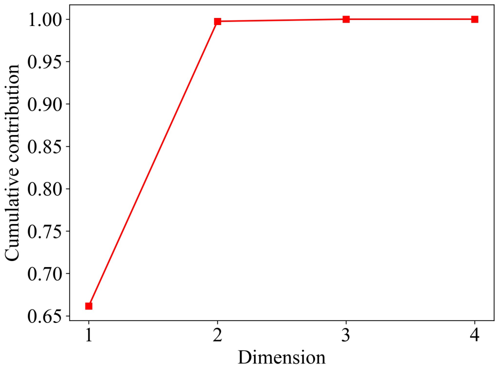


Fig. 5 Cumulative contribution statistics of each parameter

The correlation analysis of the compressed features shows that the correlation between different variables is zero, which greatly removes the redundant features, as seen in Fig. 6.

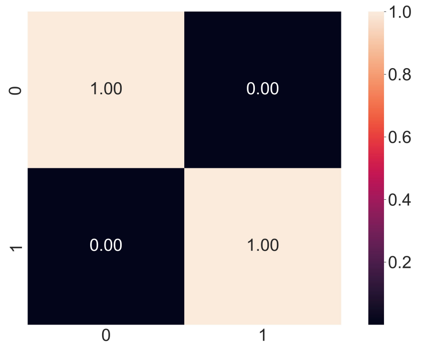


Fig. 6 The correlation coefficients of the compressed features

V. Ground Extraction Method Based on Self-Organizing Maps

The data after PCA is the condensed feature of the original point cloud, which still cannot directly distinguish whether the sample belongs to the ground. Self-organizing map [16] is an unsupervised artificial neural network that optimizes itself through competitive learning. The network maintains the topology of the input relies on the competition between neurons and the nearest neighbor relation function, and extracts features while eliminating the problem of over-dependence on data sets in most neural networks. The paper introduces SOM to roughly cluster the dimension-reduced 2D data and feature classification, which can efficiently achieve the final ground extraction.

A. Principle of SOM

SOM consists of an input layer and a competing layer (output layer), where the input layer organizes the input data. The dimensionality of the input vector determines the number of neurons. The competition layer completes the competition, collaboration and adaptation of the model. Competition means that when the data is fed into the competition layer, the model calculates the similarity between the sample α_i and each node to select the node with the highest similarity as the winner node. Collaboration implies that the neighborhood radius determines the winning neighborhood of the winning node, and the neighborhood function updates the weights of the neighborhood nodes. Adaptation makes the winning node and the neighboring nodes more sensitive to the specific input value by the activation function and the update of node weights. Typically, the adjustment size of the adaptation weights is controlled by the learning rate, which decays with the learning time and plays a decreasing role in the convergence rate of the model. Fig. 7 is the network structure of SOM.

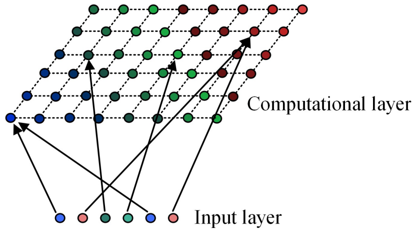


Fig. 7 The network structure of SOM

B. The ground extraction of point cloud based on SOM

According to the characteristics of the compressed data, the neurons' number of the input layer is two, and the competitive layer is 30×30 (obtained by experiment). The similarity is calculated according to the Euclidean distance. The neighborhood is delimited by a triangular region with a radius of 2 from the winning point. Learning rate (the initial value is 0.5), neighborhood radius (the initial value is 2) and the weights of neurons (set the initial value randomly) would decay gradually with iterating, and they can be obtained as:

$$lr(t)=\frac{0.5}{1+\frac{t}{T}} \tag{13}$$

$$r(t)=\frac{2}{1+\frac{t}{T}} \tag{14}$$

$$\omega(t)=\omega(t-1)+\frac{0.5}{1+3\frac{t}{T}+2\left(\frac{t}{T}\right)^2}(\omega_{win}(t)-\omega(t-1)) \tag{15}$$

where T is the total number of samples (similar to the epochs in supervised learning), t denotes the current training count, $lr(t)$, $r(t)$, $w(t)$, and $w_{in}(t)$ are the learning rate, the neighborhood radius, the weight of the updated nodes and the weight of the winner node of the t-th sample respectively.

After obtaining the 30×30 mapping results generated by the SOM during testing, the labels of a frame point cloud of the training set are used to annotate classification for the obtained mapping nodes and complete the separation of ground and other points. The model is tested with 23201 labeled data from the first 11 sequences of SemanticKitti to validate the performance of the proposed method. Each sequence is divided into the training set and the test set according to 6:4.

The paper uses *precision*(P), *recall*(R), *F1* and *accuracy*(acc) to describe the model performance, which as formulated in (16) to (19):

$$P=\frac{n_{TP}}{n_{TP}+n_{FP}} \tag{16}$$

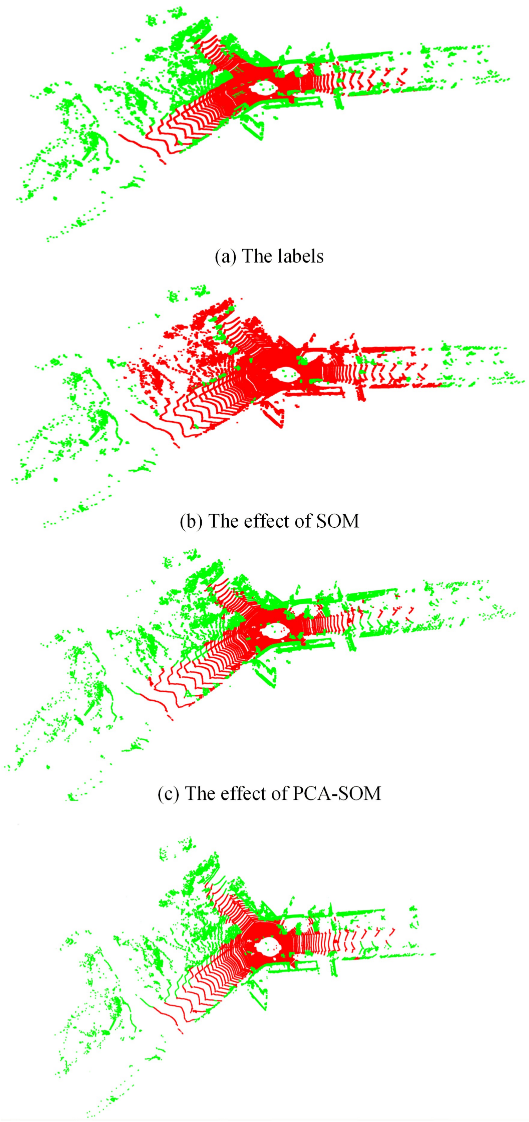
$$R=\frac{n_{TP}}{n_{TP}+n_{FN}} \tag{17}$$

$$F1=2\times\frac{P\times R}{P+R} \tag{18}$$

$$acc=\frac{n_{TP}+n_{TN}}{n_{TP}+n_{FP}+_{TN}+_{FN}} \tag{19}$$

where n_{TP} represents the number of positive samples predicted to be positive, n_{FP} represents the number of negative samples predicted to be positive, n_{TN} represents the number of negative samples predicted to be positive, and n_{FN} represents the number of positive samples predicted to be negative.

Precision describes how many positive samples in the prediction result are truly positive samples; *recall* describes how many positive samples are detected in the prediction. *F1* is a weighted average of *precision* and *recall*, which can make an efficient evaluation of the detection results for datasets with the uneven class distribution. *Accuracy* is more suitable for the detection of datasets with balanced category distribution. TABLE II. shows the detection parameters of different models, and Fig. 8 shows the detection results of the corresponding models. It can be seen that the basic SOM network can basically complete the ground extraction at sparse point clouds, but the recognition ability at dense places is poor; the introduction of PCA dimensionality reduction not only effectively improves the real-time ground extraction but also further improves the extraction ability of the model at dense point clouds, which is because PCA reduces the feature dimensionality, as well as enhances the data discrepancy through orthogonal transformation to reduce the difficulty of feature identification; the addition of the improved statistical outlier removal filter algorithm can further improve the detection accuracy and make the final pavement extraction effect meet the practical application.



(a) The labels

(b) The effect of SOM

(c) The effect of PCA-SOM

(d) The effect of filter-PCA-SOM

Fig. 8 The detection results

TABLE II. THE DETECTION RESULTS

Model	*Labels*	*Precision*	*Recall*	*F1*	*Accuracy*	*Time/s*
SOM	Ground	0.86	0.75	0.80	0.81	1.337
	Others	0.77	0.87	0.82		
PCA-SOM	Ground	0.96	0.91	0.94	0.94	0.876
	Others	0.91	0.96	0.94		
Filter-PCA-SOM	Ground	0.97	0.93	0.95	0.95	1.036
	Others	0.93	0.97	0.95		

VI. CONCLUSION

With improved performance and reduced cost, LiDAR has become a common perception for various scenarios. The collected point cloud data is commonly used for environmental perception and high-precision mapping. However, data redundancy and high computational pressure on the system are currently limiting the further application of point clouds. The paper analyzes the effectiveness of different information in the original point cloud and selects the briefest feature representation to filter, reduce dimension and extract ground points, which lays the foundation for further extraction of object features in the scene. Compared with SOM and PCA-SOM, the proposed method can improve the accuracy and speed of ground extraction. In the future, we will conduct research on the perception of driving objects on the road.

REFERENCES

[1] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens*., vol. 59, no. 01, pp. 85-101, May. 2004.

[2] J. Susaki, "Adaptive slope filtering of airborne LiDAR data in urban areas for digital terrain model (DTM) generation," *Remote Sens*, vol. 04, pp. 1804-1819, Jun. 2012.

[3] X. Lin and J. Zhang, "Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments," *Remote Sens*, vol. 06, pp. 1294-1326, Feb. 2014.

[4] P. Axelsson, "DEM generation from laser scanner data using adaptive TIN models," *ISPRS J. Photogramm. Remote Sens*., vol. 33, pp. 110-117, 2000.

[5] J. Chen, "Practice of airborne LiDAR point cloud filtering method based on triangulated irregular network," in *2014 IEEE Workshop Adv. Res. and Technol. Industry Appl.(WARTIA)*, Sep. 2014, pp. 1284-1286.

[6] C. Chen, M. Wang, B. Chang, and Y. Li, "Multi-Level Interpolation-Based filter for airborne LiDAR point clouds in forested areas," *IEEE Access*, vol. 8, pp. 41000-41012, Feb. 2020.

[7] E. Bayram, P. Frossard, E. Vural, and A. Alatan, "Analysis of airborne LiDAR point clouds with spectral graph filtering," *IEEE Geosci. Remote Sens. Lett*, vol. 15, no. 8, pp. 1284-1288, Aug. 2018.

[8] B. Zou, H. Qiu, and Y. Lu, "Point cloud reduction and denoising based on optimized downsampling and bilateral filtering," *IEEE Access*, vol. 8, pp. 136316-136326, Jul. 2020.

[9] Y. Ren, T. Li, J. Xu, W. Hong, Y. Zheng, and B. Fu, "Overall filtering algorithm for multiscale noise removal from point cloud data," *IEEE Access*, vol. 9, pp. 110723-110734, Jul. 2021.

[10] N. Li, S. Yue, and B. Jiang, "Adaptive and feature-preserving bilateral filters for three-dimensional models," *Traitement Du Signal*, vol. 37, no. 2, pp. 157–168, Apr. 2020.

[11] M. Elhoseny and K. Shankar, "Optimal bilateral filter and convolutional neural network based denoising method of medical image measurements," *Measurement*, vol. 143, pp. 125–135, Sep. 2019.

[12] Y. Budisusanto, M. N. Cahyadi, I. W. Farid, M. R. Ubaidillah, and D. W. Imani, "Low cost LiDAR prototype design for 3D papping," in *2021 Int. Conf. Adv. Mechatronics, Intell. Manufacture Ind. Automat. (ICAMIMIA)*, Dec. 2021, pp. 13-17.

[13] H. Ferdowsi, S. Jagannathan, and M. Zawodniok, "An online outlier identification and removal scheme for improving fault detection performance," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 5, pp. 908-919, May. 2014.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, et al. "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *2019 IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, Nov. 2019, pp. 9296-9306.

[15] A. Seghouane, N. Shokouhi, and I. Koch, "Sparse principal component analysis with preserved sparsity pattern," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3274-3285, July. 2019.

[16] S. Aly and S. Almotairi, "Deep convolutional Self-Organizing Map network for robust handwritten digit recognition," *IEEE Access*, vol. 8, pp. 107035-107045, Jun. 2020.