

# Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction<sup>☆</sup>

Yao Duan<sup>a</sup>, Chuanchuan Yang<sup>a,\*</sup>, Hao Chen<sup>b</sup>, Weizhen Yan<sup>b</sup>, Hongbin Li<sup>a</sup>

<sup>a</sup> State Key Laboratory of Advanced Optical Communication Systems and Networks, Peking University, Beijing 100871, China

<sup>b</sup> SenseFuture Technologies Co., Ltd, Beijing 100025, China

## ARTICLE INFO

### Keywords:

LiDAR  
Point cloud  
Noise reduction  
Principal component analysis  
Clustering

## ABSTRACT

Signals emitted by LiDAR sensors would often be negatively influenced during transmission by rain, fog, dust, atmospheric particles, scattering of light and other influencing factors, causing noises in point cloud images. To address this problem, this paper develops a new noise reduction method to filter LiDAR point clouds, i.e. an adaptive clustering method based on principal component analysis (PCA). Different from the traditional filtering methods that directly process three-dimension (3D) point cloud data, the proposed method uses dimension reduction to generate two-dimension (2D) data by extracting the first principal component and the second principal component of the original data with little information attrition. In the 2D space spanned by two principal components, the generated 2D data are clustered for noise reduction before being restored into 3D. Through dimension reduction and the clustering of the generated 2D data, this method derives low computational complexity, effectively removing noises while retaining details of environmental features. Compared with traditional filtering algorithms, the proposed method has higher precision and recall. Experimental results show a F-score as high as 0.92 with complexity reduced by 50% compared with traditional density-based clustering method.

## 1. Introduction

LiDAR is a high-precision sensor used to measure the location and shape of objects and form high-quality 3D point cloud images [1, 2], which has been widely applied to autonomous driving [3,4], 3D reconstruction [5–7], industrial measurement [8,9] and many other areas [10].

Due to adverse effects of atmospheric particles, multipath returns caused by diffuse reflection [11], as well as adverse weather conditions such as rain and snow [12], the point cloud image obtained by LiDAR sensors suffers a lot of noises, which fall into three main categories: isolated outliers, clustered noises, and noise points near signals. In order to obtain high-quality point cloud images, these noises have to be removed.

To this end, many point cloud denoising methods have been developed. Rusu et al. proposed statistical outlier removal (SOR) filter which differentiates signals from noises by calculating the average distance between each point and its nearest K points [11]. Jiang and his colleagues proposed the spatial frequency (SF) outlier filter which constructs a ball with each point as the center and removes the noise points based on the number of points within the ball [13]. The above-mentioned statistical outlier removal filters cannot identify all outliers,

and the radius outlier removal filters may falsely remove a lot of useful environmental features [12]. The cell histogram filter proposed by Carrilho et al. [11] and a dynamic radius outlier removal filter proposed by Charron et al. [12] were intended to address these problems. But the performance of radius outlier removal filter was only tested in a moderately snowy setting. Wang also proposed an adaptive ellipsoid searching filter based on the previous radius filter [14]. The center of the ellipsoid is the target point to be detected. The noises are identified based on the number of nearby points in the ellipsoid. However, this algorithm is too complex to have ideal real-time performance.

Traditional filtering algorithms are often limited to isolated outliers. In the areas close to the sensor, noise points tend to gather together, and traditional distance-based filtering algorithms are not applicable in such scenarios. Meanwhile, in the areas far away from the sensor, the signal points are often so sparse that important data on environmental features often fail to be retained by traditional methods, resulting in low precision and recall of the overall point cloud. Therefore, it is necessary to introduce a machine learning method to remove noise points that cannot be filtered by traditional methods.

<sup>☆</sup> This work was supported by National Key R&D Program of China under Grant No. 2019YFB1802904.

\* Corresponding author.

E-mail addresses: [duan.yao@pku.edu.cn](mailto:duan.yao@pku.edu.cn) (Y. Duan), [yangchuanchuan@pku.edu.cn](mailto:yangchuanchuan@pku.edu.cn) (C. Yang), [chenhao@sensefuture.cn](mailto:chenhao@sensefuture.cn) (H. Chen), [yanweizhen@sensefuture.cn](mailto:yanweizhen@sensefuture.cn) (W. Yan), [lihb@pku.edu.cn](mailto:lihb@pku.edu.cn) (H. Li).

Clustering is a machine learning technique that can be either supervised or unsupervised. The density-based spatial clustering algorithm is widely used to denoise 3D point cloud images [15,16].

Ni et al. proposed a point cloud filtering algorithm combining clustering and iterative graph cuts to process the point cloud data captured by airborne lidar [17]. Kim et al. proposed a graph-based spatial clustering algorithm for better segmentation of point clouds while reducing the background noise of each cluster [18]. And density-based spatial clustering of applications with noise (DBSCAN) is a widely used clustering technique, which is very effective at noise-reduction [19]. The problem is that all the three methods above are often accompanied by too high complexity to fulfill real-time tasks [17–19].

In addition, deep learning methods can also be used for point cloud denoising [20,21]. D. Zhang et al. proposed a point-wise learning architecture that consists of an encoder and a decoder [20]. D. Lu et al. proposed a normal estimation method for point cloud filtering while preserved deep features [21].

In the case of large-scale point cloud data, segmentation may be required [22–24]. Different segmentation methods have been explored. A. Vo et al. proposed a region-growing algorithm for the fast surface patch segmentation of 3D point clouds in urban settings [22]. And L. Landrieu et al. proposed a graph-structured deep metric learning approach to over-segment point cloud [24].

Different from all the filtering methods talked above which directly process 3D point cloud data, this paper proposes a new noise-reduction method which can generate low algorithm complexity through dimension reduction based on principal component analysis (PCA), while retaining environmental features. We refer to the proposed method as the PCA-based adaptive clustering filtering method (PCAAC). The innovation points of this method are as follows:

- (1) The proposed method employs PCA technology for dimension reduction (3D→2D) and generates 2D data by extracting the first principal component and the second principal component of the original data with little information attrition. Since main signal processing is executed on 2D data, this method can significantly reduce the overall complexity while effectively removing noises and preserving environmental features.
- (2) We propose an adaptive clustering method for 2D point cloud before they are restored into 3D. Through adaptive parameter setting on 2D space spanned by two principal components, the precision and recall of point cloud images are significantly improved.
- (3) To address the near-far effect caused by distance differences from the LiDAR sensor to scan points, the proposed method adopts a different region segmentation approach, which can help yield better noise-reduction effects.

Compared with the density-based clustering algorithm, the method proposed in this paper has higher accuracy and recall, with the F-score increased to 0.92 and the complexity reduced by as much as 50%. Compared with the traditional filtering algorithms such as the radius outlier removal filter, the proposed method has higher performance, with the F-score up by 50% and the complexity reduced by three times.

The rest of this paper is organized as follows: Section 2 lays out the principle of the proposed method. Section 3 analyzes the complexity of each filtering algorithm. Section 4 describes the filtering performance. Finally, conclusions are drawn.

## 2. Low-complexity point cloud filtering by PCA based dimension reduction

### 2.1. PCA based dimension reduction for LiDAR

PCA is a technology that can extract main feature components of data [25,26]. Robust PCA is a matrix decomposition algorithm, which is

less sensitive to noise and can handle high-dimensional image data [27–29]. In the PCA method, the eigenvector corresponding to the smallest eigenvalue is often noises related and discarding this eigenvector can help with noise-reduction. Assuming that the number of 3D data points is  $m$ , these data points are first arranged into matrix  $\xi$  with 3 rows and  $m$  columns. Each row of matrix  $\xi$  is zero-centered by subtracting the mean  $\bar{\xi}$ , which can ensure that the average value of each row is zero.

Then the covariance matrix  $C$  is obtained by

$$C = \frac{1}{m} \xi \xi^T. \quad (1)$$

After that, three eigenvectors are arranged into matrix  $E = (e_1, e_2, e_3)$  and the following equation must be satisfied:

$$E^T C E = \Lambda \quad (2)$$

where  $\Lambda$  is a diagonal matrix. Therefore, three eigenvectors and their corresponding eigenvalues are identified. The eigenvectors are arranged into a matrix in rows with decreasing eigenvalues, and the first two rows of eigenvectors are taken to generate matrix  $P$ . The third row contains little information and is thus removed. The original 3D data can be converted into 2D as:

$$Y_{2 \times m} = P_{2 \times 3} \times \xi_{3 \times m}. \quad (3)$$

There are altogether  $m$  new 2D data points, each composed of only the first principal component and the second principal component. Converting the point cloud data from 3D to 2D can greatly reduce computational complexity of subsequent signal processing algorithms.

Based on this theory, this paper proposes an adaptive clustering filtering method. Different from the traditional filtering methods that directly process 3D point cloud data, the proposed method converts the original 3D data into 2D through PCA and then clusters the generated 2D data for noise reduction before restoring the data into 3D.

### 2.2. Adaptive clustering filtering method based on PCA

The proposed PCA-based clustering filtering method is outlined in Fig. 1. The original data are first divided into region segments in response to the decreasing density of point cloud further away from the sensor. PCA is then carried out on each region for dimension reduction following the method in Section 2.1. The first principal component and the second principal component, whose accumulative variance contribution rate exceeds 95%, are extracted and designated as  $X$ -axis and  $Y$ -axis coordinates. Next, the 2D clustering algorithm is used to cluster the planar point clouds. The numbers of point clouds in each cluster are calculated and compared with the threshold to identify and remove noise points. Afterwards, the filtered 2D point cloud data are restored into 3D point cloud data. Finally, the restored 3D data for all segments will be stitched together to form a complete point cloud image. Each step will be elaborated in the following sections.

#### 2.2.1. Region segmentation approach

The distance from each area to the sensor is different, and the number of buildings and street scenes is also varied, requiring different filtering parameters. By segmentation, data points in different areas will be filtered with different parameter settings based on their respective environmental features.

Fig. 2(a) shows the traditional segmentation method with the LiDAR sensor at the center. The whole space is divided into multiple cube regions with a same volume. However, the distances from the points within the same cube to the sensor are different, and the noise distribution is also different. For example,  $d_1 \neq d_2 \neq d_3$  and  $d_4 \neq d_5$ , which goes against the principle of segmentation. Therefore, we propose an improved segmentation approach, as shown in Fig. 2(b).

Given that real-life environment is mainly composed of buildings and streets with limited heights,  $Z$ -values in Cartesian coordinate system usually range between 0 and 30 m. While the height of an object

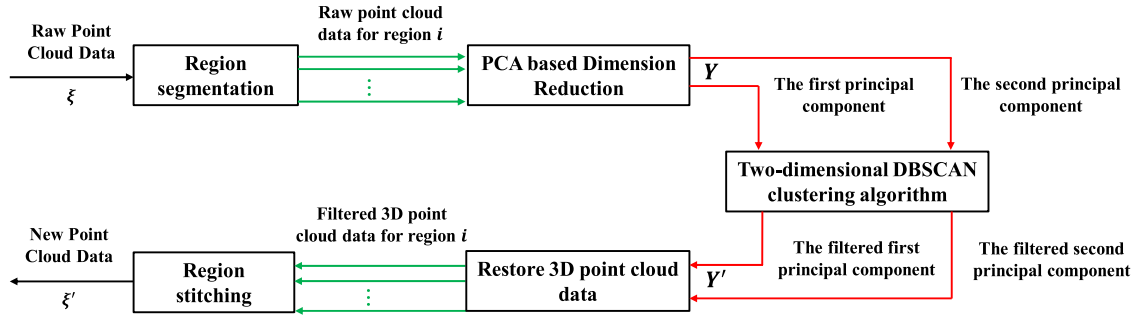


Fig. 1. The block diagram of the proposed PCAAC method.

is within a certain range, the ground (X–Y plane) is theoretically indefinite. Therefore, we propose a cylinder-structure segmentation approach.

As shown in Fig. 2(b), with the LiDAR sensor at the center, the whole space is segmented into many cylinder blocks, and the space sections between each cylinder are the segment regions of point cloud.

In this cylinder model, to have a same volume for each region, the bottom area between every two neighboring cylinders should satisfy

$$\pi r_1^2 h = \pi r_2^2 h - \pi r_1^2 h = \dots = \pi r_i^2 h - \pi r_{i-1}^2 h \quad (4)$$

where  $r_i$  ( $i = 1, 2, \dots, t$ ) is the radius of the  $i$ th cylinder, that is, the distance from the point on the cylinder to the LiDAR sensor (center), and  $h$  is the height of cylinder. This formula can be simplified into  $r_i = \sqrt{i} r_1$ , meaning when the bottom radius of cylinder  $i$  is  $\sqrt{i}$  times of that of cylinder 1, the volume of each segment is equal.  $r_1$  is determined by the point furthest away from the sensor as

$$r_1 = \frac{r_{max}}{\sqrt{t}} \quad (5)$$

After segmentation, PCA is performed on the raw 3D data to extract main components and convert 3D data into 2D data.

### 2.2.2. 2D-DBSCAN clustering

Based on the theory of dimension reduction in Section 2.1, a 2D-DBSCAN clustering algorithm is proposed to filter 2D point cloud on a plane. By calculating the number of point clouds in each cluster and comparing it with the threshold, we can identify noises. A cluster with fewer point clouds than the threshold is identified as a noise cluster; otherwise, it is identified as a signal cluster.

The original 3D point cloud is represented by an unordered set of points

$$\xi = \left\{ (x_i, y_i, z_i)^T, i = 1, 2, \dots, m \right\} \quad (6)$$

where  $(x_i, y_i, z_i)^T \in R^3$  is a coordinate vector and  $m$  is the total number of points.

According to formula (3), the original 3D data can be converted into 2D data after PCA dimension reduction. Each 2D point cloud is composed of the first principal component and the second principal component, represented by  $f$  and  $s$  respectively.

Therefore, the point cloud input into DBSCAN cluster can be expressed as

$$Y = \left\{ (f_i, s_i)^T, i = 1, 2, \dots, m \right\} \quad (7)$$

where  $(f_i, s_i)^T \in R^2$  is a coordinate vector in the 2D space spanned by two principal components.

The 2D-DBSCAN algorithm is displayed in Algorithm 1.

$Y$  refers to the data set composed of the first principal component  $f$  and the second principal component  $s$ . Specifically, in the 2D plane, the distance  $d$  can be calculated as:

$$d = \sqrt{(f_1 - f_2)^2 + (s_1 - s_2)^2} \quad (8)$$

For every unvisited point  $p_i$  in data set  $Y$ , identify all points within its  $\epsilon$  neighborhood with *RANGE\_QUERY* function and gather these points into a subset  $N_i$ . The number of points within subset  $N_i$  is referred to as  $|N_i|$ . If  $|N_i| < minpts$ , point  $p_i$  is labeled as an outlier; if  $|N_i| \geq minpts$ , add all points in subset  $N_i$  to cluster  $S$  ( $S \leftarrow N_i$ ).

For an unvisited point  $p_j$  in subset  $N_i$ , gather all points within its  $\epsilon$  neighborhood into a subset  $N_j$ . If  $|N_j| \geq minpts$ ,  $N_j$  will be added to  $S$  ( $S \leftarrow S \cup N_j$ ).

---

#### Algorithm 1 2D-DBSCAN Algorithm

---

**Input:**  $Y, \epsilon, minpts, d, \psi$

**Data:** label

**Function:** *RANGE\_QUERY*( $\cdot$ )

```

1: for each point  $p_i = (f_i, s_i)^T$  in  $Y$  do
2:   if label $_{p_i}$  = undefined then
3:     Neighbors  $N_i \leftarrow RANGE\_QUERY(Y, d, p_i, \epsilon)$ 
4:     if  $|N_i| < minpts$  then
5:       label $_{p_i} \leftarrow Noise$ 
6:     else
7:        $S \leftarrow new\ cluster$ 
8:        $S \leftarrow N_i$ 
9:     end if
10:  end if
11:  for each point  $p_j = (f_j, s_j)^T$  in  $N_i$  do
12:    if label $_{p_j}$  = undefined then
13:      Neighbors  $N_j \leftarrow RANGE\_QUERY(Y, d, p_j, \epsilon)$ 
14:      if  $|N_j| \geq minpts$  then
15:         $S \leftarrow S \cup N_j$ 
16:      end if
17:    end if
18:  end for
19: end for
20: Output:  $\{S_1, S_2, \dots, S_k\}$ 
21: for each cluster  $S_i$  in data set  $\{S_1, S_2, \dots, S_k\}$ 
22:   if  $|S_i| < \psi$  then
23:     for each point  $p_c = (f_c, s_c)^T$  in  $S_i$  do
24:       label $_{p_c} \leftarrow Noise$ 
25:     end for
26:   end if
27: end for

```

---

The above process is repeated until there is no unvisited point, resulting in  $k$  clusters:  $\{S_1, S_2, \dots, S_k\}$ . The number of data points within cluster  $S_i$  ( $i = 1, 2, \dots, k$ ) is referred to as  $|S_i|$ . Compare  $|S_i|$  with the threshold  $\psi$ . If  $|S_i| > \psi$ , cluster  $S_i$  is a signal cluster. Otherwise, it is identified as a noise cluster and removed. The threshold  $\psi$  of each area is different, depending on the distance from the LiDAR sensor. The farther away from the sensor, the larger the signal block obtained by clustering, and thus the larger the threshold should be correspondingly.

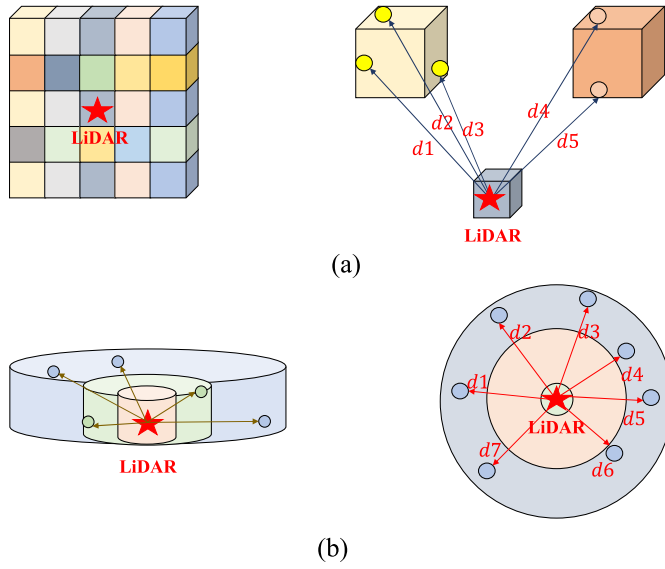
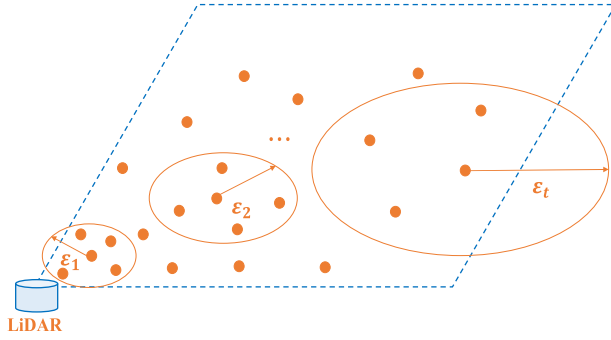


Fig. 2. Traditional (a) and improved (b) region segmentation methods.

Fig. 3. Adaptive  $\epsilon$  parameters for each region segmentation.

After removing the noises, we can get the filtered point cloud data  $Y'$ , which can be expressed as:

$$Y' = \left\{ (f_l, s_l)^T, l = 1, 2, \dots, n \right\} \quad (9)$$

where  $(f_l, s_l)^T \in R^2$  is the filtered coordinate vector in the 2D space and  $n$  is the number of points remaining after filtering.

After being denoised, the 2D data is restored into 3D. The denoised 3D data can be obtained through:

$$\xi' = P^T Y' + \bar{\xi} \quad (10)$$

where  $Y'$  is the filtered point cloud data set in 2D space and  $\bar{\xi}$  is the mean subtracted from the original matrix  $\xi$ .

### 2.3. Key parameters

There are three key parameters in the proposed PCAAC method, i.e.  $\epsilon$ ,  $minpts$  and  $\psi$ . For each region segmentation, the three parameters are adaptively modified to obtain the best clustering effect and high-precision point cloud data.

Considering that the density of point cloud declines further away from the sensor, the  $\epsilon$  for each region segment is modified adaptively. As shown in Fig. 3, the closer the point cloud is to the LiDAR sensor, the higher the density is, and the smaller the  $\epsilon$  value should be.

$\epsilon$  for a certain region can be defined as:

$$\epsilon_i = kr_i \quad (11)$$

Table 1

The complexity analysis of different methods.

| Filter                             | Complexity (+)  | Complexity (×)  |
|------------------------------------|-----------------|-----------------|
| Statistical outlier removal filter | $O(m^2 \log m)$ | $O(m^2 \log m)$ |
| Radius outlier removal filter      | $O(9m^2)$       | $O(3m^2)$       |
| Density-based clustering method    | $O(13m^2)$      | $O(1.5m^2)$     |
| PCAAC method                       | $O(10m^2)$      | $O(m^2)$        |

where  $\epsilon_i$  ( $i = 1, 2, \dots, t$ ) refers to the neighborhood radius for region  $i$ .  $r_i$  refers to the distance between the point cloud in region  $i$  and the LiDAR sensor. Since  $r_i = \sqrt{i}r_1$ , there is:

$$\epsilon_i = \sqrt{i}\epsilon_1 \quad (12)$$

where  $\epsilon_1$  is equal to 1.

The default value of  $minpts$  is set as 10. Based on  $\epsilon$  and  $minpts$ , silhouette value is calculated. The value of  $minpts$  for each region is adjusted until the largest silhouette value is obtained.  $\psi$  is used to distinguish noise clusters from signal clusters. Given that the number of point clouds for a general object is more than 100, the default value of  $\psi$  is set at 100. The threshold range is between 50 and 200.

### 3. Complexity analysis

Table 1 compares complexities of different point cloud filtering methods. Compared with the statistical outlier removal filter, the complexity of the proposed PCAAC method is reduced by an order of magnitude. Compared with the radius outlier removal filter, the complexity of PCAAC method is reduced by three times. Compared with the existing density-based clustering method, the complexity of PCAAC method is reduced by about 50%. The complexities listed in Table 1 are calculated when the corresponding filtering methods yield the best performance and their performances are analyzed in the following Section.

### 4. Performance analysis

The original point cloud data used in performance analysis came from global public database *Open Topography*. The point cloud processing software is MATLAB provided by MathWorks.

Fig. 4(a) shows the original point cloud data, including buildings, grasslands, trees, street lamps and other environmental features. There are a total of 480,000 point clouds with a density of 0.94 pts/m<sup>3</sup>. The X coordinate range is [−80 m, 80 m], the Y range is [−80 m, 80 m], and the Z range is [0 m, 20 m]. Fig. 4(b) shows the filtering results by the proposed PCAAC filter. In the process of point cloud filtering, low-density sampling points such as street lamps, telegraph poles, power lines and building walls would often be falsely removed by traditional filtering algorithms as noise points, greatly compromising the precision and recall. This problem can be solved by the proposed PCAAC method.

Take a certain region in the original point cloud image as an example. As shown in Fig. 5(a), this image contains 32,839 points with a density of 2.74 pts/m<sup>3</sup>, including two buildings, a street lamp in the middle, a large grassland, a tree and scattered noise points. Fig. 5(b) shows the filtering result of the statistical outlier removal filter. It can remove outliers but unable to remove noise clusters. Fig. 5(c) shows the filtering result of the radius outlier removal filter. Though it can remove both outliers and noise clusters, it also damaged the environmental features by falsely removing the street lamp in the middle as noises. Fig. 5(d) shows the filtering result of the proposed PCAAC method. With the new method, most noise points were removed with details in environmental features preserved.

In order to evaluate the effectiveness of the filter, four basic parameters and five performance evaluation indicators were used. The four basic parameters are *true positive (TP)*, *false positive (FP)*, *true negative (TN)* and *false negative (FN)* [1]. The five performance evaluation



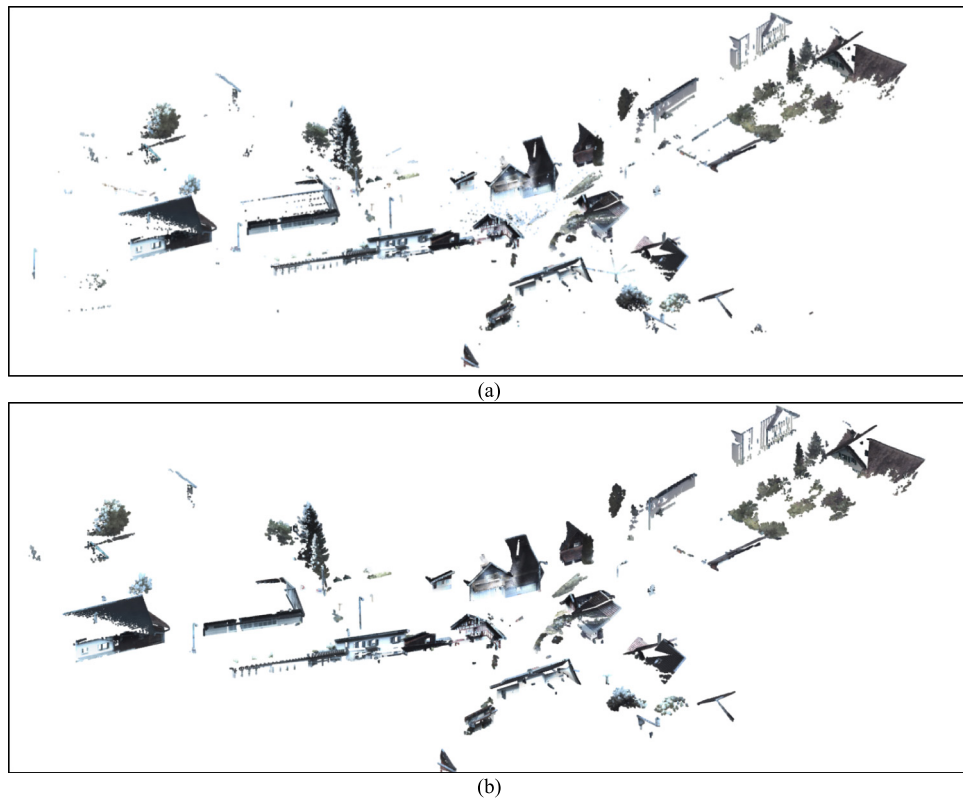


Fig. 4. (a) Original point cloud image, (b) filtered point cloud image by the proposed PCAAC filter.

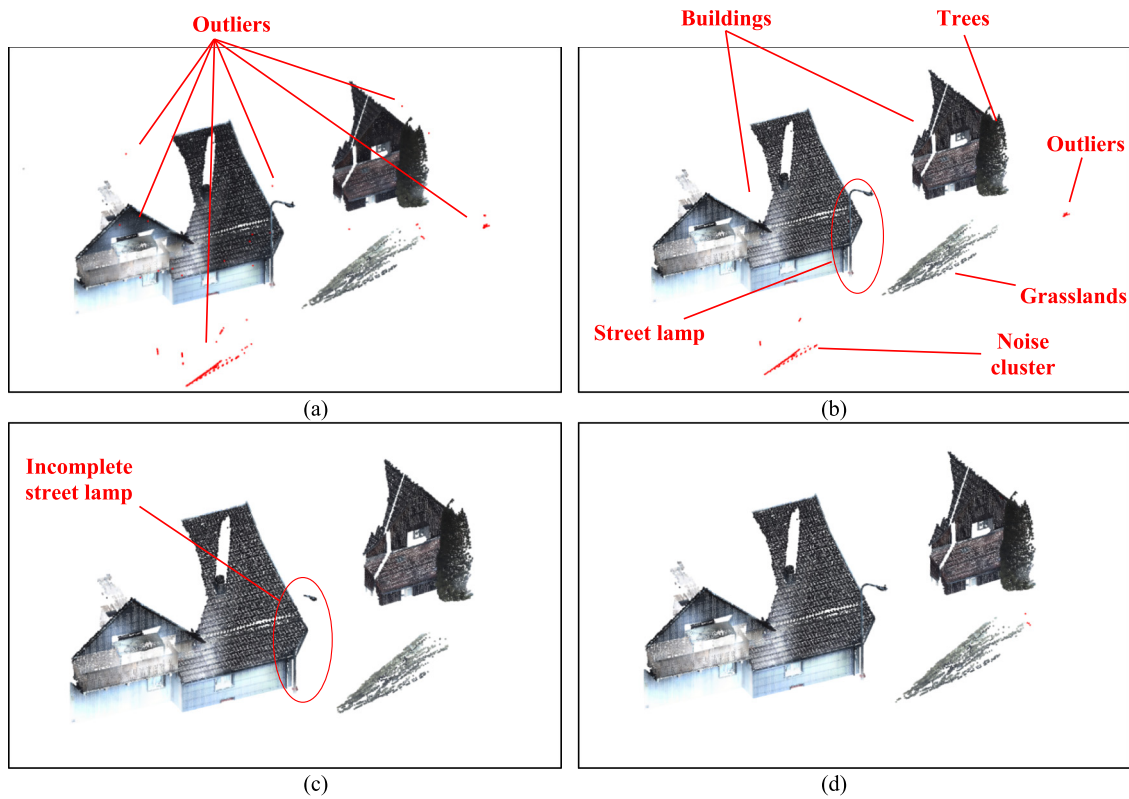


Fig. 5. Original point cloud image and denoised images by different algorithms. (a) Original point cloud image, (b) filtered image by statistical outlier removal filter, (c) filtered image by radius outlier removal filter and (d) filtered image by the proposed PCAAC filter.

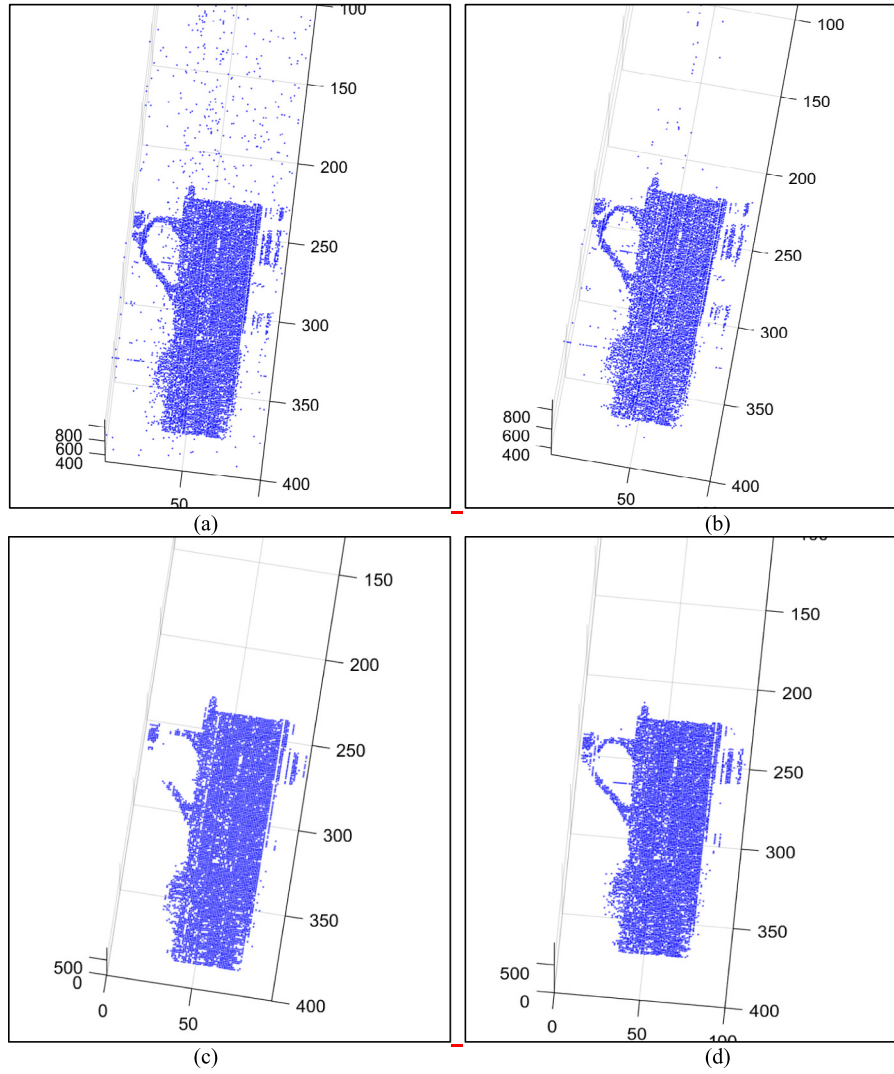


Fig. 6. (a) Original point cloud image, (b) 3D point cloud image filtered by statistical outlier removal filter, (c) 3D point cloud image filtered by radius outlier removal filter and (d) 3D point cloud image filtered by the proposed PCAAC filter.

indicators are *accuracy*, *error*, *precision*, *recall* and  $F_1$  value, which are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (13)$$

$$Error = \frac{FP + FN}{TP + FP + TN + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

Ground truth is manually calibrated. The five performance indicators for different filtering methods are compared in Table 2.

The statistical outlier removal filter is not effective in removing noise clusters, resulting in a very low recall rate. The two-stage statistical outlier removal filter has a higher recall rate, but its  $F_1$  is still outperformed by the radius outlier removal filter. The main problem of the radius outlier removal filter is that useful environmental features (such as street lamps) was removed as noise points, resulting in very low precision and low  $F_1$  value. Density-based clustering method yielded significantly higher precision and  $F_1$  value than traditional methods. It combines the advantages of statistical outlier removal filter and radius outlier removal filter in removing noise points while retaining

Table 2

Performance evaluation of different methods.

| Filter                                       | Accuracy | Error | Precision | Recall | $F_1$ |
|--|----------|-------|-----------|--------|-------|
| Statistical Outlier Removal Filter           | 99.42%   | 0.58% | 58.82%    | 28.85% | 0.39  |
| Two-stage Statistical Outlier Removal Filter | 99.52%   | 0.48% | 64.29%    | 56.25% | 0.60  |
| Radius Outlier Removal Filter                | 99.28%   | 0.72% | 46.95%    | 100%   | 0.64  |
| Density-based clustering method              | 99.68%   | 0.32% | 68.03%    | 96.15% | 0.80  |
| PCAAC method                                 | 99.89%   | 0.11% | 97.27%    | 86.00% | 0.92  |

useful environmental features. The proposed PCAAC method yielded better outcomes than the existing density-based clustering method. It features high precision and recall rate, and its  $F_1$  value reached 0.92. Compared with statistical outlier removal filter, two-stage statistical outlier removal filter, radius outlier removal filter and density-based clustering method,  $F_1$  value of the PCAAC method was increased by 135%, 53%, 43% and 15% respectively.

To further test the performance of the proposed method, an experiment was conducted via 3D LiDAR imaging using single-photon

**Table 3**

Performance evaluation of different methods.

| Filter                                       | Accuracy | Error  | Precision | Recall | F <sub>1</sub> |
|--|----------|--------|-----------|--------|----------------|
| Statistical Outlier Removal Filter           | 87.94%   | 12.06% | 74.64%    | 35.32% | 0.48           |
| Two-stage Statistical Outlier Removal Filter | 92.14%   | 7.86%  | 100%      | 50.07% | 0.67           |
| Radius Outlier Removal Filter                | 91.45%   | 8.55%  | 64.79%    | 100%   | 0.78           |
| Density-based clustering method              | 94.42%   | 5.58%  | 86.18%    | 76.90% | 0.81           |
| PCAAC method                                 | 96.70%   | 3.30%  | 95.62%    | 82.86% | 0.88           |

avalanche diode (SPAD) detector. By using the light source with wavelength of 850 nm and the latest SPAD detector with a total of 400×100 pixels, depth imaging and intensity imaging were performed.

Fig. 6(a) shows the original point cloud image derived from the experiment with a significantly higher level of noise compared with Fig. 5(a). Fig. 6(b) shows the filtering result of the statistical outlier removal filter, Fig. 6(c) shows the filtering result of the radius outlier removal filter and Fig. 6(d) shows the filtering result of the proposed PCAAC method in this article. The statistical outlier removal filter did not perform well in removing outliers and radius outlier removal filter mistakenly removed some of object features (cup handle). With the proposed PCAAC method, most noise points were removed and details in object features were preserved.

The five performance indicators for different filtering methods are compared in Table 3. Results from different methods are compared, and the performance of the proposed PCAAC method was overall satisfactory. When the original point cloud image is sparse and noisy, F<sub>1</sub> value was still maintained at as high as 0.88.

## 5. Conclusions

Point cloud filtering is one of the key steps during lidar signal processing, since noise reduction is essential to obtaining high-quality point cloud data. An adaptive clustering filtering algorithm based on principal component analysis is proposed in this paper. Compared with previous methods, the proposed PCAAC method has higher precision, recall and F<sub>1</sub> value even for highly noisy point cloud. Compared with statistical outlier removal filter and radius outlier removal filter, the F-score of the proposed PCAAC filter increased by 135% and 43% respectively with complexity reduced respectively by over 10 times and three times. Compared with the density-based clustering method, the F-score of the proposed PCAAC filter increased by 15% with complexity lowered by 50%. Most noises can be removed while details in environmental features are retained, yielding satisfactory point cloud images. With dimension reduction through PCA, the computational complexity and the running time of the algorithm can be significantly reduced, ensuring better real-time performances.

## 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.

## References

- [1] X.T. Nguyen, et al., A gradient-aware line sampling algorithm for LiDAR scanners, *IEEE Sens. J.* (2020) 1.
- [2] B. Nagy, et al., 3D CNN-Based semantic labeling approach for mobile laser scanning data, *IEEE Sens. J.* 19 (21) (2019) 10034–10045.
- [3] P. Vines, et al., High performance planar germanium-on-silicon single-photon avalanche diode detectors, *Nature Commun.* 10 (1) (2019) 1086–1089.
- [4] Z. Huang, et al., Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding, *IEEE Sens. J.* (2020) 1.
- [5] J. Tachella, et al., Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers, *Nature Commun.* 10 (1) (2019) 4984–4986.
- [6] J. Tachella, et al., Bayesian 3D Reconstruction of complex scenes from single-photon lidar data, *SIAM J. Imaging Sci.* 12 (1) (2019) 521–550.
- [7] K. Kuzmenko, et al., 3D LIDAR Imaging using Ge-on-Si single-photon avalanche diode detectors, *Opt. Express* 28 (2) (2020) 1330.
- [8] B. Schwarz, LIDAR MApping the world in 3D, *Nat. Photonics* 4 (7) (2010) 429–430.
- [9] L. Huo, et al., Supervised spatial classification of multispectral LiDAR data in urban areas, *PLoS One* 13 (10) (2018) e0206185.
- [10] Y. Altmann, et al., Spectral unmixing of multispectral lidar signals, *IEEE Trans. Signal Process.* 63 (20) (2015) 5525–5534.
- [11] A.C. Carrilho, et al., Statistical outlier detection method for airborne lidar data, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-1* (2018) 87–92.
- [12] N. Charron, et al., De-noising of lidar point clouds corrupted by snowfall, 2018, <http://dx.doi.org/10.1109/CRV.2018>.
- [13] J. Jia-Jia, et al., An effective frequency-spatial filter method to restrain the interferences for active sensors gain and phase errors calibration, *IEEE Sens. J.* 16 (21) (2016) 7713–7719.
- [14] X. Wang, et al., An adaptive ellipsoid searching filter for airborne single-photon lidar, *IEEE Geosci. Remote Sens. Lett.* 14 (8) (2017) 1258–1262.
- [15] J. Ren, et al., LANDMARC Indoor positioning algorithm based on density-based spatial clustering of applications with noise-genetic algorithm-radial basis function neural network, *Int. J. Distrib. Sens. Netw.* 16 (2) (2020) 155014772090783.
- [16] P. Rodríguez-González, et al., Mobile LiDAR system: New possibilities for the documentation and dissemination of large cultural heritage sites, *Remote Sens.* 9 (3) (2017) 189.
- [17] H. Ni, et al., Joint clusters and iterative graph cuts for ALS point cloud filtering, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 11 (3) (2018) 990–1004.
- [18] J. Kim, et al., Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3D point clouds, *Sensors* 19 (18) (2019) 3926.
- [19] E. Schubert, et al., DBSCAN Revisited: Why and how you should (still) use DBSCAN, *ACM Trans. Database Syst.* 42 (3) (2017) 1–21.
- [20] D. Zhang, et al., Pointfilter: Point cloud filtering via Encoder–Decoder modeling, 2020, arXiv preprint.
- [21] D. Lu, et al., Deep feature-preserving normal estimation for point cloud filtering, *Comput. Aided Des.* 125 (2020) 102860.
- [22] A. Vo, et al., Octree-based region growing for point cloud segmentation, *ISPRS J. Photogramm. Remote Sens.* 104 (2015) 88–100.
- [23] Y. Xu, et al., A voxel- and graph-based strategy for segmenting man-made infrastructures using perceptual grouping laws: Comparison and evaluation, *Photogramm. Eng. Remote Sens.* 84 (6) (2018) 377–391.
- [24] L. Landrieu, M. Boussaha, Point Cloud Over segmentation with Graph-Structured Deep Metric Learning, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7440–7449.
- [25] Z. Xu, Z. Tan, C. Yang, Eigenvector basis expansion-based phase noise suppression method for CO-OFDM systems, *IEEE Photonics Technol. Lett.* 29 (13) (2017) 1124–1127.
- [26] M. Hayman, R.A. Stillwell, S.M. Spuler, Fast computation of absorption spectra for lidar data processing using principal component analysis, *Opt. Lett.* 44 (8) (2019) 1900.
- [27] O. Solomon, et al., Deep unfolded robust PCA with application to clutter suppression in ultrasound, *IEEE Trans. Med. Imaging* 39 (4) (2020) 1051–1063.
- [28] J. Liu, B.D. Rao, Sparse Bayesian learning for robust PCA: Algorithms and analyses, *IEEE Trans. Signal Process.* 67 (22) (2019) 5837–5849.
- [29] F. Wen, et al., Robust PCA using generalized nonconvex regularization, *IEEE Trans. Circuits Syst. Video Technol.* 30 (6) (2020) 1497–1510.