Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 54–61

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

A 3D Point Cloud Filtering Algorithm based on Surface Variation Factor Classification

Chao Chuan Jia[a,b], Chuan Jiang Wang[a], Ting Yang[b], Bing Hui Fan[*]*[a], Fu Gui He[b]

[a] College of Mechanical & Electronic Engineering, Shandong University of Science and Technology, Qingdao, Shandong, 266510,China.
[b] College of Electronics and Information Engineering, West Anhui University, Lu'an, Anhui, 237012, China.

Abstract

In order to effectively smooth the noise in 3D point cloud without losing the detailed features of the model, a new filtering algorithm based on surface variation factor segmentation is proposed. The method adopts different filtering algorithms on different feature regions of the model. Firstly, the normal vectors of a point cloud model are estimated by using weighted principal component analysis (PCA) method, and the surface variation factor of each point is estimated. Secondly, the point cloud model is divided into flat regions and mutant regions by comparing the surface variation factor of the sampling point with the average surface variant factor of the sampling point k-neighborhood. Finally, the improved median filtering algorithm is applied to flat regions, and improved bilateral filtering algorithm is applied to mutant regions. Experimental results show that it has a preferable smoothing effect and reserves the detail features of point cloud.

Keywords: Weighted Principal Component Analysis; Surface Variation Factor; Improved Median Filtering; Adaptive Bilateral Filtering.

1. Introduction

In recent years, with the promotion of computer technology and 3D scanning equipment, 3D reconstruction of real objects has become a research hotspot in the field of computer vision. 3D reconstruction technology is widely used in reverse engineering, 3D printing, virtual reality, archaeology, medicine and other fields [1-3]. 3D point cloud modeling is an effective objects modeling method, however, the premise of reconstructing the scanned object

[*] * Corresponding author. Tel.: +(86)18860453758
Email: fanbh58@163.com/ccjia@hfcas.ac.cn

10.1016/j.procs.2019.06.010

is to obtain the real data of the object surface. But because of some human or environmental factors, as well the defects of the scanning device itself, unreasonable noise will inevitably exist in the scanning data, however, these noise data will cause serious problems for subsequent related processing in modeling and measurement [4-6]. Therefore, 3D point cloud filtering is a key step before modeling. The purpose of filtering is to effectively eliminate and smooth the noise in 3D point cloud model, and reserve the original detail features of the object surface.

2. Related work

Many kinds of point cloud filtering algorithms have been proposed by domestic and foreign scholars in recent years. Some representative algorithms will be reviewed and discussed in this section. For a more detailed and complete overview, you can refer to the relevant review literature [7]. Zeng et al. [8] is inspired by the thought of Gauss-Seidel iteration, the point cloud filtering is carried out by using the interpolation method combining Lagrange operator with local surface fitting, this filtering algorithm may not achieve the ideal filtering effect for the point cloud data with non-uniform distribution for the reason that the fitting effect of quadratic least squares surface is not ideal. Lin et al.[9] proposed a scatter point cloud filtering algorithm based on the parameter adaptive anisotropic gaussian kernel function. According to the local distribution characteristics of point cloud, this algorithm can adaptively adjust the attenuation speed of each filtering main direction, and can effectively maintain the original sharp features of point cloud model while realizing the noise elimination of scattered point cloud. however, a lot of feature information are calculated to reserve the sharp features, thus this algorithm has high computational complexity. Among the commonly used denoising algorithms, the bilateral filtering algorithm [10] which is used widely can achieve better denoising effect, however, it is easy to produce local over fairing problem when processing large scale noise and cannot effectively maintain the detailed features of point cloud model. Li et al.[11] put forward a denoising algorithm for point cloud based on noise classification. The large-scale noise is removed by statistical filtering and radius filtering, then the small-scale noise is smoothed with fast bilateral filtering, this algorithm can effectively maintain the geometric features of the scanned object, however, the correlation statistics parameters and radius parameters will have a serious impact on the filtering effect. Wu et al. [12] designed a filtering algorithm based on the average curvature feature classification, traditional median and bilateral filtering algorithm are applied to different feature regions respectively. But since the sampling point is not moving in the normal direction, the vertices will drift. Moorfield B. et al. [13] modified the normal vectors of point cloud according to the bilateral filtering operator, and then improved bilateral filtering algorithm based the modified normal vectors is applied to the whole point cloud. This algorithm can preserve edge features and other small detail features, but it is easy to produce over fairing and feature shrinkage when removing the large-scale noise in point cloud. Cao et al. [14] classified the noise of point cloud into feature points and non-feature points, and then two different bilateral filtering factors are used to smooth the different feature points. This algorithm has good retention effect on the features of the model, but its time cost is high. Zheng et al. [15] proposed rolling normal filtering algorithm for point clouds. This algorithm applies the rolling guidance filter to modify iteratively the normal vectors of the point cloud, and then positions of point cloud are updated by iterative optimization of the weighted normal and weighted position energy, it can successfully remove different scales features from the point cloud. However, it needs to be iterated many times, so it has a large amount of computation. Gu et al.[16] divided the point cloud into two different feature regions according to the Gaussian curvature feature, then different denoising smoothing strategies are applied to the different feature regions separately, it is capable of simply and effectively to smooth the noise and reserve the detail features of the model, however, Gaussian curvature feature which is second-order curvature information is sensitive to noise, thus there will be a lot of misclassification when point cloud is classified.

In order to solve the problem of vertices drifting and local surface over fairing in the above existing filtering algorithms, a new filtering algorithm based on surface variation factor segmentation is proposed in this paper. We divide the point cloud model into two different regions according the surface variation factor, and the normal vectors are estimated by using the robust PCA, then different filtering algorithms are applied to different regions. The improved median filtering algorithm is applied to flat regions, and adaptive bilateral filtering algorithm is applied to mutant regions. The result of filtering shows that this method has a preferable smoothing effect and reserve the detail features of point cloud.

3. Proposed Algorithm

Firstly, the normal vectors are estimated by using weighted PCA and surface variation factors of point cloud are estimated; secondly, the point cloud is divided into two different regions by comparing the surface variation factor of the sampling point with the average surface variation factor of the k-neighborhood points; finally, two different filtering algorithms which based on the new normal vectors of point cloud are adopted to smooth the two different regions.

3.1. Estimating the normal vectors based robust PCA

Where, d_i and d_j respectively denote the distances between p_i , p_j in the k-neighborhood points and neighborhood mean $\overline{p}$. Then the weighted mean $\overline{p}_w$ is calculated according to Eq. (3). The weighted covariance matrix will be calculated according to Eq. (4).Normal vector is the basis of computing all kinds of feature information of point cloud. At present, the common method for estimating the normal vector of point cloud is the principal component analysis based on the local neighborhood points covariance analysis. Suppose that the sampling point p and its k-neighborhood $N(p)$ are given in the point cloud, then the covariance matrix C will be constructed as follows:

$$C=\frac{1}{k}\sum_{i=1}^{k}(p_i-\overline{p})(p_i-\overline{p})^T \ (p_i\in N(p))$$
$$\overline{p}=\frac{1}{k}\sum_{i=1}^{k}p_i \tag{1}$$

Where $\overline{p}$ represents the neighborhood mean of the k-neighborhood. It is well known that the degree of k-neighborhood points deviation from the neighborhood mean $\overline{p}$ can be described by the covariance matrix C . Since that the covariance matrix C is a 3×3 symmetric semi-positive definite matric, its three eigenvalues are non-negative real numbers, and the corresponding three eigenvectors are orthogonal to each other. Assume that $\lambda_0,\lambda_1,\lambda_2$ denote the three eigenvalues, e_0,e_1,e_2 represent the corresponding three eigenvectors, if $\lambda_0\le\lambda_1\le\lambda_2$, then the normal vector n of the sampling point p can be approximated by the corresponding eigenvector e_0 of λ_0 . Although principal component analysis as a statistical method has a certain anti-noise effect, when the noise in the point cloud is relatively large-scale, the method is often unable to satisfy the need. Therefore, a weighted PCA method is applied to improve the robustness of the traditional PCA. The weights of the points in neighborhood of the sampling point p are assigned according to the distances between them and the neighborhood mean $\overline{p}$. A small weight is assigned to a point which is far from the neighborhood mean, on the contrary, a large weight is assigned to a point which is relatively close to the neighborhood mean. The specific formula is shown in Eq. (2). This weights distribution method effectively reduces the influence of outliers on the neighborhood mean, the principal components of covariance matrix will be not biased by these noises in point cloud.

$$w_i=\begin{cases}1, & d_i=0\\ \dfrac{1}{d_i\cdot\sum_{j=1}^{k}d_j}, & d_i\neq0\end{cases} \tag{2}$$

Where, d_i and d_j respectively denote the distances between p_i , p_j in the k-neighborhood points and neighborhood mean $\overline{p}$. Then the weighted mean $\overline{p}_w$ is calculated according to Eq. (3). The weighted covariance matrix will be calculated according to Eq. (4).

$$\overline{p}_w=\frac{\sum_{j=1}^{k}w_ip_i}{\sum_{j=1}^{k}w_i} \tag{3}$$

$$C_w = \frac{1}{k}\sum_{i=1}^{k}\sqrt{w_i}\cdot(p_i - \bar{p}_w)\cdot(p_i - \bar{p}_w)^T \tag{4}$$

Once the weighted covariance matrix C_w is estimated, three eigenvalues and the corresponding three eigenvectors of C_w are calculated. the robust normal vector n of the sampling point p is obtained.

3.2. Dividing different regions based on surface variation factor

In order to smooth noise information in 3D point cloud model, and reserve the detail features while removing the noise. 3D point cloud is divided into two different regions by using the surface variant factor of the sampling point in this paper, and two different denoising methods are applied to different regions. We compared the surface variant factor of the sampling point with the average surface variant factor of the sampling point k-neighborhood and then the whole point cloud is divided into flat region and mutant region. Assume that the sampling point p and its k-neighborhood points are given, then the surface variation factor of the sampling point p is constructed as follows:

$$\sigma(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{5}$$

Where the λ_0 , λ_1 , λ_2 denote the three eigenvalues of matrix C . Obviously, since that $\sigma(p)$ is the first-order differential information, it is not as insensitive to noise as second-order curvature information [17]. Surface variation factor can describe how closely the local k-neighborhood points of the sampling point form a smooth planar patch. If the surface variation factor approaches zero, it means that the local neighborhood forms a smooth plane. In general, the surface variation factor is relatively small in the smooth region of low curvature and low noise, whereas the surface variation factor is large in high curvature, high noise level and sharp feature regions [18].Let the k-neighborhood average surface variation factor is:

$$\overline{\sigma(p)} = \frac{1}{k}\sum_{i=1}^{k}\sigma(p_i) \tag{6}$$

If the surface variation factor $\sigma(p)$ of the sampling point p is less than $\overline{\sigma(p)}$, then the sampling point p is classified as flat regions, otherwise, if the $\sigma(p)$ is greater than $\overline{\sigma(p)}$, then the sampling point p is classified as mutant regions.

3.3. Denoising and smoothing different regions respectively

Two different filtering algorithms are applied to the different feature regions, in this part. Median filtering is a nonlinear signal processing technique based on the ranking statistics theory. The basic principle of median filtering is to replace the value of a point in the digital image or digital sequence with the median value of the k-neighborhood of the sampling point, so that the surrounding pixel value is close to the real value, thus it has good filtering effect on impulse noise, in particular, it can effectively reserve the detail features of the signal while removing noise. Therefore, improved median filtering method is used to smooth the flat regions. Bilateral filtering algorithm is a nonlinear filter by combining Gaussian filter weighting function with the feature preserving weighting function, which can maintain the detailed features of the point cloud model while smoothing the noise. Thus, improved bilateral filtering algorithm is applied to smooth the mutant regions.

3.3.1 Improved median filtering algorithm

Since that the surface variation factor of the points change slightly in the flat regions, so applying median filtering algorithm [19] can obtain ideal result. However, when the traditional median filtering algorithm is used to smooth the 3D point cloud, a tridimensional window is usually set to slide on the point cloud, and then the points in the window are sorted according to the values of z-coordinate. Finally, the point corresponding to the middle point of the sorted sequence is used to replace the center point of the window. Obviously, the traditional median algorithm will drift the vertex position, so it is difficult to obtain the ideal filtering effect. In order to improve the vertex drift of traditional median filtering algorithm, an improved median filtering algorithm is designed.

Assume that the sampling point p and its k-neighborhood $N(p)$, the normal vector n of the sampling point are given in the point cloud. All points within $N(p)$ are projected to the normal vector n, and the values of the projection are calculated and sorted. The central point of the projection sequence is regarded as the median point $proj_{mid}$, and the median point is taken as the distance of the sampling point p moving along the normal vector n. The new position p_{new} of sampling point p after moving is defined as follow:

$$p_{new} = p + proj_{mid} \bullet \tag{7}$$

3.3.2 Improved bilateral filtering algorithm

Bilateral filtering algorithm is a common method of 2D image filtering, which has been extended to 3D point cloud filtering. 3D bilateral filtering algorithm is to achieve the goal of removing the noise by moving the sampling point containing noise along the normal vector director of the sampling point. Therefore, the specific form of the improved bilateral filtering algorithm based on the normal vectors is as follow:

$$p_{new} = p + \alpha \bullet \tag{8}$$

$$\alpha = \frac{\sum_{p_i \in N(p)} w_c(\| p - p_i \|) w_s(\| \langle n, n_i \rangle - 1 \|) \langle p - p_i, n \rangle}{\sum_{p_i \in N(p)} w_c(\| p - p_i \|) w_s(\| \langle n, n_i \rangle - 1 \|)} \tag{9}$$

Where, α is the bilateral filtering factor, n represents the normal vector of the sampling point p, p_i and n_i denote the k-neighborhood point and the corresponding normal vector of point p_i, respectively, p_{new} denotes the new position after moving. w_c is the spatial domain Gaussian filtering weight function and w_s is the feature retention weight function, their expression can be defined as below.

$$w_c(x) = e^{-\frac{x^2}{2\sigma_c^2}} \tag{10}$$

$$w_s(x) = e^{-\frac{x^2}{2\sigma_s^2}} \tag{11}$$

Where, σ_c and σ_s are Gaussian filtering parameters. σ_c denotes the influence factor of the distance from the sampling point to its k-neighborhood points on the sampling point, which is used to control the degree of smoothing. The large the σ_c is, the more neighborhood points are selected, and the smoother the 3D point cloud model is. σ_s represents the influence factor of the projection of the distance from the sampling point p to its neighborhood points p_i in its normal direction, which is used to control the degree of feature retention. The large the σ_s is, the longer distance that the sampling point moves in its normal direction, the better the feature retention of model is. Generally, σ_c takes the radius of k-neighborhood of the sampling point p, σ_s is equal to the projection values standard deviation of k nearest-neighborhood on the new vector n' of the sampling point p. Their specific expressions are as follow:

$$\sigma_c = \max(\|p - p_i\|), i \in [1, k] \tag{12}$$

$$\sigma_s = \sqrt{\frac{1}{k-1} \sum_{i=1}^{k} (\varepsilon_i - \bar{\varepsilon})^2} \tag{13}$$

Where, $\varepsilon_i = \langle p - p_i, n \rangle$, $\bar{\varepsilon} = \frac{1}{k} \sum_{i=1}^{k} \varepsilon_i$.

In equation (9), the traditional bilateral filtering factor is improved, and the cosine value of the normal inclined angle between the sampling point and the local neighborhood point is chosen as the feature retention weight factor of the bilateral filtering factor. Since that the normal change of the sampling point can reflect the change of the feature of the plane in local neighborhood, when the local neighborhood is mutant region, its feature change greatly, and the corresponding normal change is also relatively large. When the normal inclined angle between the neighborhood point and the sample point is within the range of 0^o to 90^o, the cosine value decreases with the increase of the included angle, so the feature retention weight of the improved bilateral filter factor is also reduced with the increase of the included angle; when the included angle is greater than 90^o, the feature retention weight is set to 0, that is, the neighborhood point no longer has any influence on the current sampling point. Therefore, by limiting the range of the normal included angle, the feature retention weight of the improved bilateral filtering factor is limited to a more reasonable range, which can enhance the feature retention ability of bilateral filtering factor.

4. Experiments and Result Analysis

In the algorithm, first the point cloud is loaded from the PCD or TXT file, the normal vectors of the point cloud are calculated by using the weighted PCA method and the surface variation factors are calculated according to the Eq.(2)-Eq.(5), and then for every sampling point of the model, the k-neighborhood points of the sampling point are searched by using KD-Tree search method, the mean surface variation factor of the k-neighborhood and the projection values of k-neighborhood points on the new normal vector of current sampling point are calculated, separately. If the average surface variation factor of the k nearest-neighborhood is greater than or equal to the surface variation of the sampling point, then the projection values are sorted by using the bubbling method, the median value is found, the position of the current sampling point is updated according to Eq.(7). If the average surface variation factor of the k nearest-neighborhood is less than the surface variation of the sampling point, the bilateral filtering factor α is calculated according to the Eq.(9), the position of the current sampling point is updated according to Eq.(8). Finally, the filtered point cloud is saved to the PCD or TXT file after all points were dealt.

A publicly available Stanford Bunny point cloud is employed to verify the effectiveness of the proposed filtering algorithm. Firstly, the original point cloud is added with a Gaussian noise of zero mean and variance σ (σ=0.05%, σ=0.1%). Secondly, four different filtering algorithms which are median filtering algorithm, bilateral filtering algorithm, filtering algorithm in [16] and proposed filtering algorithm are applied to point cloud after adding noise. Finally, the deviation of the reconstruction model of the filtered point cloud and original point cloud is analyzed. In proposed method, only the size k of the neighborhood points needs to be set by the user. If k is set to small, the algorithm running time can be saved, but the filtering effective is very poor, if k is set to large, the algorithm is not only time-consuming but also may lead the model to over smooth. The algorithm will produce a better filtering effective when k is set to 15 in this paper. The results of experiments are shown in Fig.1 and Fig.2. Fig.1(a) and Fig.2(a) are the reconstruction models of original Bunny point cloud, Fig.1(b) and Fig.2(b) are the reconstruction models of point cloud with a Gaussian noise, Fig.1(c) and Fig.2(b) are the reconstruction models of filtered point cloud by the median filtering algorithm, Fig.1(d) and Fig.2(d) are the reconstruction models of filtered point cloud by the bilateral filtering algorithm, Fig.1(e) and Fig.2(e) are the reconstruction models of filtered point cloud by the filtering algorithm in paper[16], Fig.1(f) and Fig.2(f) are the reconstruction models of filtered point cloud by the proposed filtering algorithm. From these figures, we can clearly find that different filtering algorithms have different filtering effectiveness. The vertices drifting not only occurs in filtered point cloud by using the median filtering algorithm, but also the detail features can't be reserved. Over fairing phenomenon appears in filtered point cloud by using bilateral filtering algorithm. Although the filtering algorithm in paper [16] has a good effectiveness, the vertices drifting still appears since that the error of the estimation normal vectors. However, the surfaces of filtered point cloud by using the proposed algorithm are not only smooth, but also the detail features are well reserved in this paper.

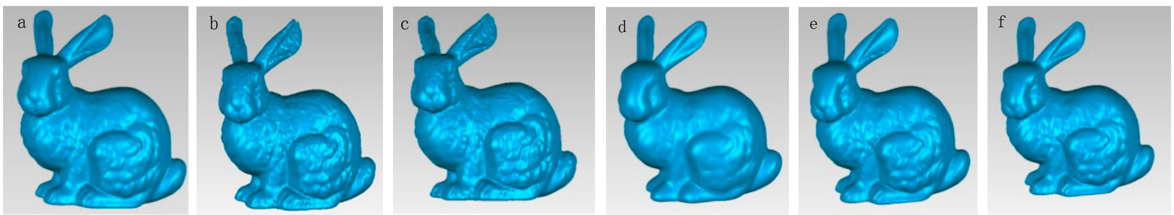

Fig.1 (a) Original model Denoising;(b) After adding noise model;(c) Median filtering algorithm; (d) Bilateral filtering algorithm; (e) Paper[16] filtering algorithm; (f) proposed filtering algorithm ($\sigma = 0.05\%$)

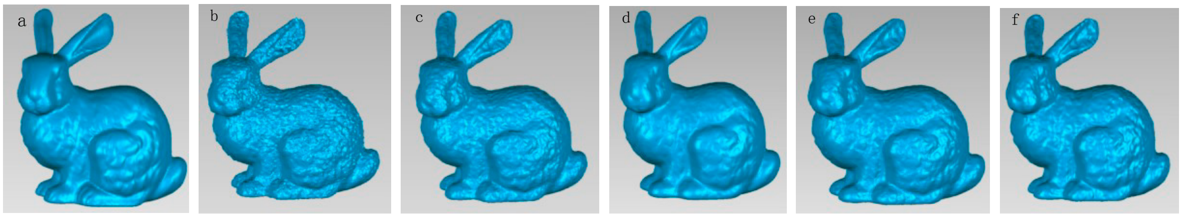

Fig.2 (a) Original model Denoising;(b) After adding noise model;(c) Median filtering algorithm; (d) Bilateral filtering algorithm; (e) Paper[16] filtering algorithm; (f) proposed filtering algorithm ($\sigma = 0.1\%$)

5. Conclusion

In this paper, in order to smooth the noise in point cloud, while the detailed features are preserved, two different filtering algorithms are applied to the different feature regions. which are classified by comparing the surface variation factor to the average surface variation factor of k-neighborhood points. First, the weighted PCA method is used to estimate the normal vectors of the point cloud model, and the surface variation factor of every sampling point is calculated, and then the point cloud is divided into flat regions and mutant regions by comparing the surface variation factor to the average surface variation factor of k-neighborhood points, finally, the improved median filtering algorithm is applied to flat regions, and adaptive bilateral filtering algorithm is applied to mutant regions. Different filtering algorithms are applied to the Stanford Bunny point cloud with a Gaussian noise (σ=0.05%, σ=0.1%). The experiment results show that the proposed algorithm can maintain detailed features and avoid vertex drift and over fairing while de-noising the point cloud.

Acknowledgement

This work supported by National Young Natural Science Foundation (NO. 61702375), China, Key Research Programs of Shandong Province (NO. 2016GSF201197), Science and Technology Plan Programs of Colleges and Universities in Shandong Province (NO. J16LB11) and Natural Science Foundation of Anhui Province, (NO. KJ2014A277).

References

1. Ma Z., Liu S., A review of 3D reconstruction techniques in civil engineering and their applications. *Advanced Engineering Informatics*, 2018, 37:163-174.
2. Intwala A. M., Magikar A., A review on process of 3D Model Reconstruction. *International Conference on Electrical, Electronics, and Optimization Techniques. IEEE*, 2016.
3. Dryanovski I., Klingensmith M., Srinivasa S. S, et al. Large-scale, real-time 3D scene reconstruction on a mobile device. *Autonomous Robots*, 2017, 41(6):1-23.
4. Wang L., Xu Y., Li Y., Aerial Lidar Point Cloud Voxelization with its 3D Ground Filtering Application. *Photogrammetric Engineering & Remote Sensing*, 2017, 83(2):95-107.
5. Han X. F., Jin J. S., Wang M. J., et al. Guided 3D point cloud filtering. *Multimedia Tools & Applications,* 2017(3):1-15.
6. Wang J. L., Chen L. J., Review on Filtering Algorithm for LIDAR Points Cloud Data. *Remote Sensing Technology & Application,* 2010, 25(5):632-638.
7. Han X. F., Jin J. S., Wang M. J., et al. A review of algorithms for filtering the 3D point cloud. *Signal Processing Image Communication,* 2017, 57:103-112.
8. Zeng F. X., Li L., Research on point cloud filtering based on Lagrange operator and surface fitting. *Journal of laser,* 2016, 37(8):75-78.
9. Lin H. B., Fu D. M., Wang Y. T. Feature preserving denoising of scattered point cloud based on parametric adaptive and anisotropic gaussian kernel. *Computer Integrated Manufacturing Systems,* 2017,23(12):2583-2592.
10. Yuan H., Pang J. P., Mo J. W., Denoising algorithm for bilateral filtered point based on noise classification. *Journal of Computer Applications,* 2015,35(8):2305-2310.
11. Li P. F., Wu H. E., Jing J. F., et al., Noise classification denoising algorithm for point cloud model. Computer Engineering and Application, 2016,52(20):188-192.
12. Wu L. S., Shi H. L., Chen H. W., Denoising of three-dimensional point data based on classification of feature information. *Optics and Precision Engineering.* 2016,24(6):1465-1473.
13. Moorfield B., Haeusler R., Klette R., Bilateral Filtering of 3D Point Clouds for Refined 3D Roadside Reconstructions. *Lecture Notes in Computer Science*, 2015, 9257:394-402.

14. Cao S., Yue J., Ma W., Bilateral filtering denoise algorithm for point cloud based on feature selection. Journal of Southeast University, 2013, 43:1-2.
15. Zheng Y., Li G., Xu X., et al. Rolling Normal Filtering for Point Clouds. *Computer Aided Geometric Design*, 2018, 62:16-28.
16. Gu X. Y., Liu Y. S., Wu Q., A filtering algorithm for scattered point cloud based on curvature features classification. *Journal of Information & Computational Science,* 2015,12(2):525-532.
17. Li W.L., Xie H., Zhang. G., et al. Adaptive Bilateral Smoothing For a Point-Sampled Blade Surface. *IEEE ASME Transactions on Mechatronics*, 2016, 21(6):2805-2816.
18. Wang Y., Feng H. Y., Outlier detection for scanned point clouds using majority voting. *Computer-Aided Design*, 2015,62:31-43.
19. Yao Z. Q., Application of median filter in point cloud data pre-processing lidar. *Laser Technology*, 2009, 33(2):213-216.