Hindawi
Journal of Sensors
Volume 2021, Article ID 7325600, 10 pages
https://doi.org/10.1155/2021/7325600

Research Article

Optimization Algorithm for Point Cloud Quality Enhancement Based on Statistical Filtering

Qian Zhao, Xiaorong Gao, Jinlong Li, and Lin Luo

School of Physical Science and Technology, Southwest Jiaotong University, Chengdu, China 610031

Correspondence should be addressed to Jinlong Li; jinlong_lee@126.com

Received 2 June 2021; Accepted 11 November 2021; Published 6 December 2021

Academic Editor: Rosalba Liguori

In the process of acquiring point cloud data by a 3D laser scanner, some problems, such as outliers, mixed points, and holes, may be caused in the target point cloud due to the external environment, the discreteness of the laser beam, and the occlusion of objects. In this paper, a point cloud quality optimization and enhancement algorithm is designed. A self-adaptive octree is established to rasterize the point cloud and calculate the density of each grid, combing with the statistical filtering to remove outliers from the point cloud data. Then, a plane projection method is used for removing the confounding points from the point cloud data. Finally, the point cloud is triangulated and a priority value is set, and then, points are preferentially inserted where the priority value is the largest to repair the holes. Experiments show that while removing outliers and confounding points, the detailed features of the point cloud can be maintained, holes are effectively filled, and the quality of the point cloud is effectively improved.

1. Introduction

In recent years, laser 3D scanning technology has developed rapidly. As a new format of data, the three-dimensional point cloud can accurately record the three-dimensional topography, geometric features, spatial coordinates, and other information about the surface of the object, and it has many advantages that two-dimensional data does not have [1]. In the process of acquiring point cloud data by a 3D laser scanner, outliers, mixed points, holes, and other defects may be generated in the target point cloud due to the external environment, the discreteness of the laser beam, and the occlusion of objects [2–5]. These defects degrade the quality of the point cloud and affect the subsequent reconstruction accuracy of the three-dimensional surface.

Pirotti et al. [6] made some assessment with the two commonly used outlier detection methods—statistical outlier removal (SOR) filter and local outlier factor (LOF) filter. Four different filtering algorithms were well used for filtering the raw point cloud data form a UAV by Mustafa Zeybek [7], and a developed methodology contributes to the reduction of errors caused by data losses in various modelling studies [8]. Fleishman et al. [9] proposed the moving least squares algorithm, in which the noise points are projected onto an estimated plane. But it is sensitive to a large number of outliers. Based on the locally optimal projection (LOP) algorithm proposed by Lipman et al. [10], Huang et al. [11] proposed weighted locally optimal projection algorithms that can remove some noise points during the projection. But these algorithms have poor robustness and cannot keep the sharp features of the point cloud well. Rusu et al. [12] proposed an algorithm based on neighborhood statistics, which needs two iterations and takes much time to process the large-scale point cloud. Meanwhile, it is a challenge to guarantee accuracy when removing the noise. Ning et al. [13] proposed a simple and effective noisy point trimming method based on two geometric feature constraints, but it is only suitable for certain types of noise. Ester et al. [14] proposed a clustering algorithm DBSCAN, which can be used for removing the redundant points including the noise points. Leal et al. [15] presented a two-step method both for normal estimation and for point position update, measuring the sparsity of sharp features while discriminating between noise and feature, but the performance in some models with more detailed features is general. In addition to removing noisy points, the optimization and enhancement of point cloud quality also need to repair the holes caused by the object occlusion. Holes repairing algorithms based on triangular meshes could be divided into two categories: surface-based methods and volume-based ones [16]. Generally, surface-based methods detect, repair, and refine the holes on the given triangular mesh [17, 18]. Volume-based methods convert the given mesh into a function with signed distance on the volume mesh to fill the holes and then extract the complete grid from the zero-order set of the distance function [19–21]. The holes repairing the algorithm proposed by Leong et al. [22] directly connect the boundary of the holes, but the repair is not effective since no new triangle surfaces are added. Liu Zhenghong and Yun [23] extracted the boundary of the holes and determined the direction of contraction by calculating the direction of the triangular surface's normal vector related to the boundary point. In this way, a complete triangular patch is iteratively generated, but the number of additional points could not be controlled. The point cloud quality optimization and enhancement algorithm designed in this paper could effectively remove the outliers and confounding points while maintaining the sharp features of the point cloud. At the same time, by setting the priority value, the number of new points could be controlled when repairing holes.

2. Theoretical Derivation

The implementation steps of the proposed method are as follows:

(1) Establishing a self-adaptive octree to rasterize the point cloud and setting the thresholds for defining the outliers to delete these points

(2) After removing the outliers, projecting the point cloud onto the local least squares fitting plane to remove the confounding points

(3) Establishing a priority value and inserting points at the positions with higher priority values until the hole filling which is completed

2.1. Outlier Removal. Outliers are usually disorganized, far away from the main point cloud, sparse, and geometrically discontinuous with inconsistent local point density. In contrast, the main point cloud is relatively concentrated and dense. The statistical filtering algorithm is based on the characteristics that the distance between the outlier points and the neighboring points is considerable while the distance between the main points and the neighboring points is small. And the statistical analysis towards the neighborhood of each point is used for removing the outliers [24]. In this paper, density is introduced based on the statistical filtering algorithm. First, an octree is established to take the minimum enclosing cube of the point cloud as the root node, and the cube is divided into eight subcubes with equal size. If a subcube contains points, it should continue to be divided into eight cubes with equal size until each point in the point cloud obtains a unique index coordinate. Then, the density of the point cloud in each small cube is calculated, and usually, the density of the subcube containing outliers is relatively small.

The specific implementation steps are as follows:

(1) Establishing a self-adaptive octree to rasterize the point cloud and calculating the density of the point cloud ρ_n in each grid

(2) For any point $p_i \in P$ of the point cloud in each grid, searching for the k-neighborhood point $p_j (j \in k)$ of p_i and calculating the distance between p_i and p_j

(3) Defining the probability of a point being an outlier as

$$\delta = \frac{\sum_{j=1}^{k} \left\| p_i - p_j \right\|}{k\rho_n} \tag{1}$$

(4) Establishing a threshold. If δ is greater than the threshold, p_i is considered an outlier and is deleted

2.2. Confounding Point Removal. Confounding points cannot be easily distinguished from the real point cloud. It is difficult for traditional algorithms to achieve a balance between removing the confounding points and maintaining the detailed and sharp features of the point cloud. In this paper, we use the least square to fit the local plane and obtain the normal vector of the local plane. The mixed points are projected onto the local fitting plane, which not only improves the quality of the point cloud but also maintains the sharp features.

The least squares method is used for fitting the local plane and obtaining the normal vector. First, we fit a local plane with a set of points composed of a point $p_i \in P$ and its k-neighborhood points. Suppose the local fitting plane is $Z = Ax + By + C$, and then, the formula of the deviation quadratic sum is as follows:

$$d = \sum_{i=1}^{k+1} \left[(Ax_i + B_i + C) - z_i \right]^2. \tag{2}$$

It can also be written in matrix form as follows:

$$d = (UB - l)^2, \tag{3}$$

where

$$U = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix},$$

$$B = \begin{pmatrix} A \\ B \\ C \end{pmatrix},$$

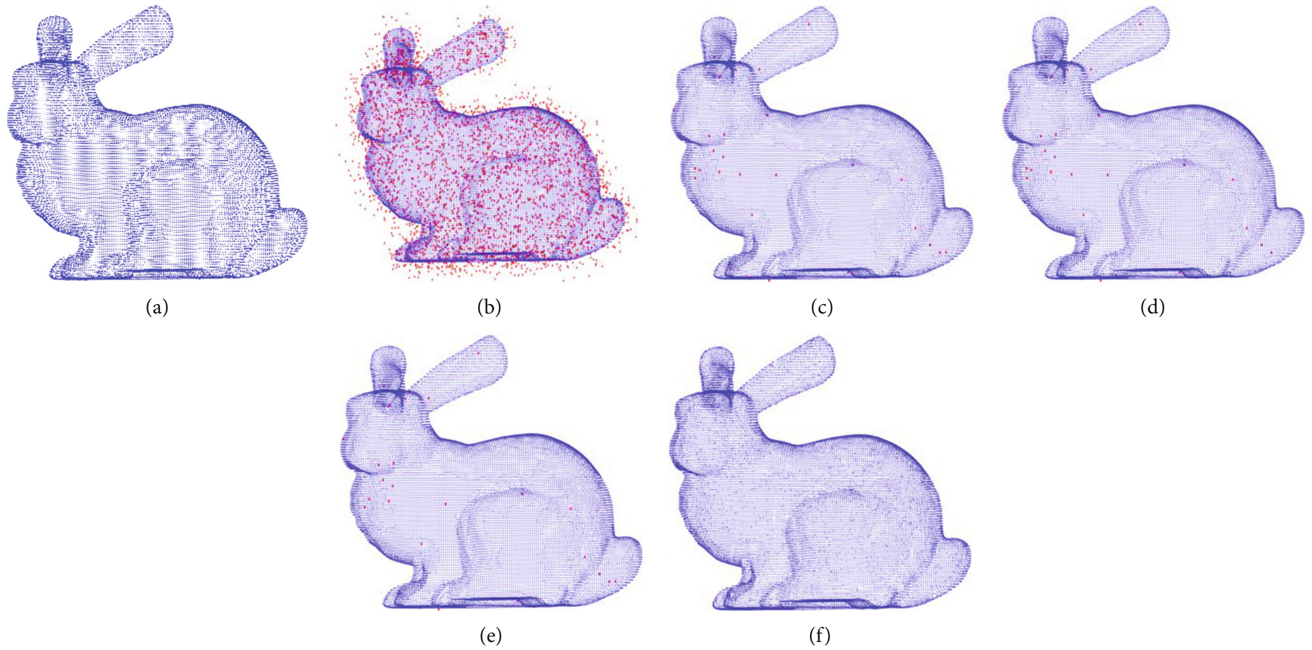


Figure 1: Denoising results of the three methods with 10% noise in the Bunny point cloud. (a) The original Bunny point cloud. (b) Bunny point cloud with 10% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.

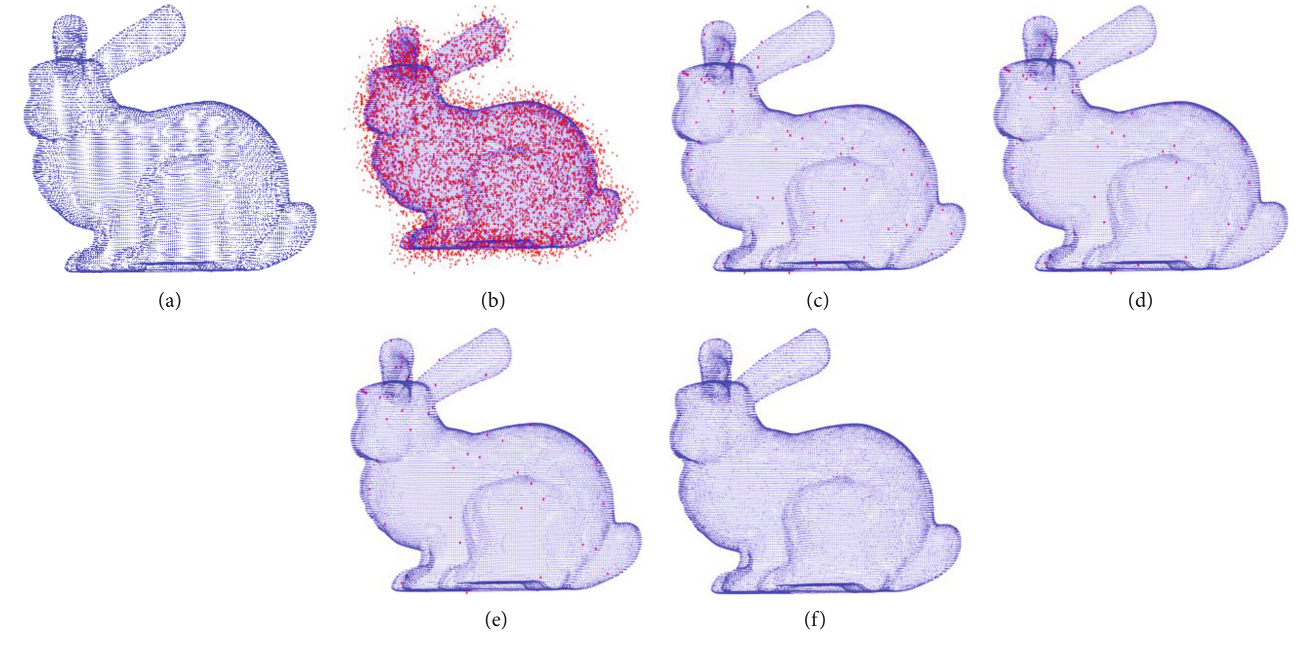
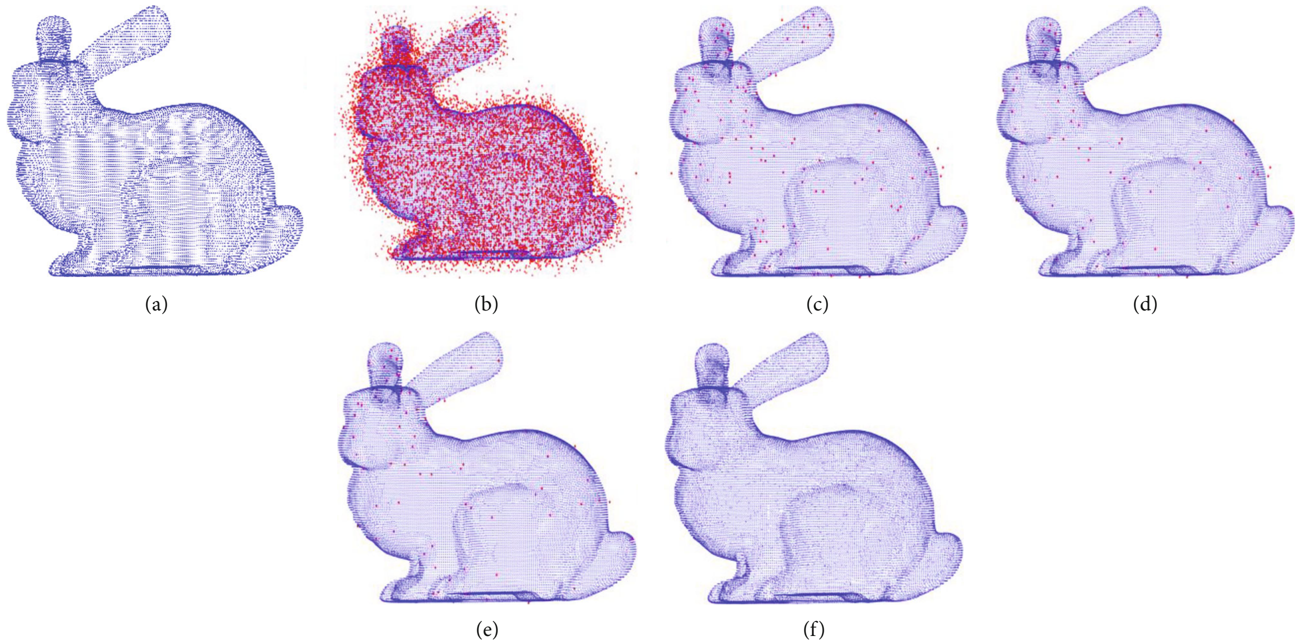


Figure 2: Denoising results of the three methods with 20% noise in the Bunny point cloud. (a) The original Bunny point cloud. (b) Bunny point cloud with 20% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.

$$l = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}. \tag{4}$$

When d takes the minimum value, the fitting degree is the best. According to the least squares criterion, we can use the following formula to obtain B.

$$B = \left(U^T U\right)^{-1} U^T l. \tag{5}$$



(a) (b) (c) (d)

(e) (f)

Figure 3: Denoising results of the three methods with 30% noise in the Bunny point cloud. (a) The original Bunny point cloud. (b) Bunny point cloud with 30% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.

Table 1: Denoising results of the three methods with 10% noise in Bunny point cloud.

Algorithms	Total number of noisy point clouds	Pd	Rd	*F*1
Statistical filtering	39542	0.6935	0.9704	0.8091
Radius filtering	39542	0.7029	0.9814	0.8189
Our method	39542	0.7541	0.7177	0.8279

Table 2: Denoising results of the three methods with 20% noise in Bunny point cloud.

Algorithms	Total number of noisy point clouds	Pd	Rd	*F*1
Statistical filtering	43173	0.6315	0.9339	0.7535
Radius filtering	43173	0.7001	0.9848	0.8184
Our method	43173	0.7441	0.9453	0.8327

Table 3: Denoising results of the three methods with 30% noise in Bunny point cloud.

Algorithms	Total number of noisy point clouds	Pd	Rd	*F*1
Statistical filtering	46732	0.5929	0.9046	0.7163
Radius filtering	46732	0.7124	0.9646	0.8192
Our method	46732	0.7588	0.9368	0.8385

In this way, the fitting plane can be obtained, and the normal vector of a certain point can also be got by substituting the coordinates of the point into the fitting plane.

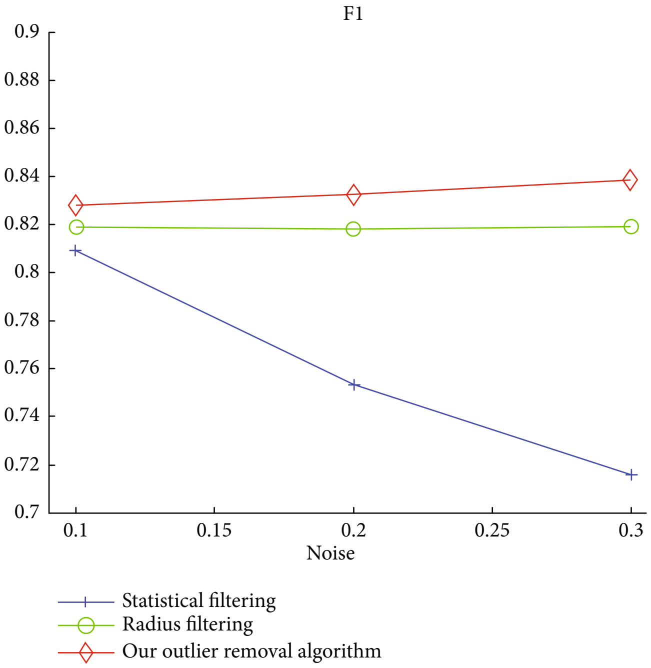


Figure 4: *F*-score of the three algorithms.

Table 4: Chamfer distance in Bunny point cloud.

Noise level	$d_{CD}(s_0, s_1)$	$d_{CD}(s_0, s_2)$	$d_{CD}(s_0, s_3)$
10%	1.5892	0.0464	0.0424
20%	2.7208	0.0748	0.0718
30%	3.7317	0.1132	0.0891

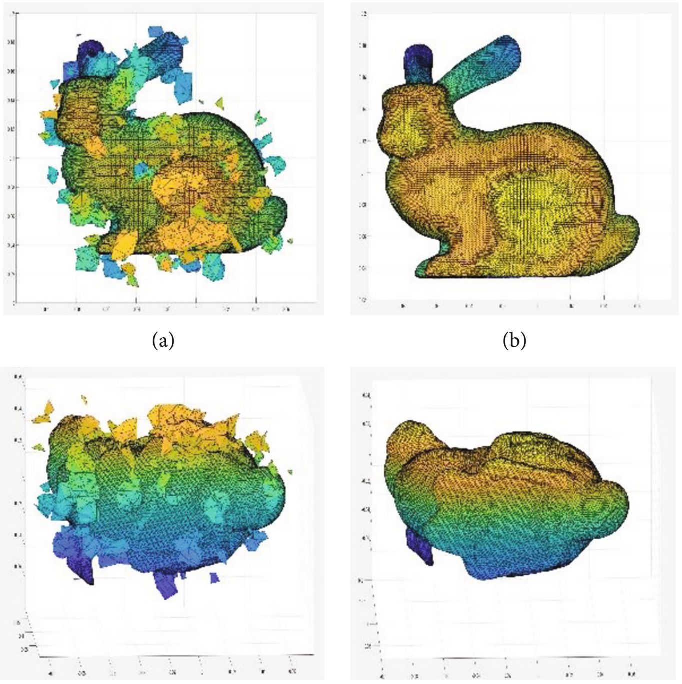
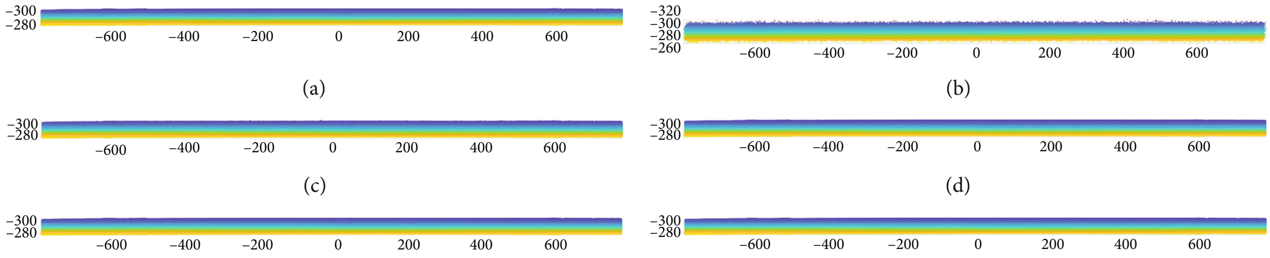

(a)
(b)
(c)
(d)

Figure 5: Triangulation effect. (a) Noisy point cloud triangulation effect. (b) Point cloud triangulation effect after the viewing angle is flipped. (d) Triangulation effect of the point cloud processed by the proposed algorithm after flipping the viewing angle.



(a)
(b)
(c)
(d)
(e)
(f)

Figure 6: Denoising results of the three methods in the pantograph model. (a) The original pantograph point cloud. (b) Pantograph point cloud with 10% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.

The specific implementation steps are as follows:

(1) For a point $p_i \in P$ in the target point cloud P, searching for its k-neighborhood points and calculating the mean value p_{mi} of the neighborhood points

(2) Using the least squares method to fit a local plane L_i and calculating the normal vector n_i of p_{mi}

(3) Calculating the projection of the vector $p_i p_{mi}$ in the normal direction to obtain the distance d from the point $p_i \in P$ to the local fitting plane L_i

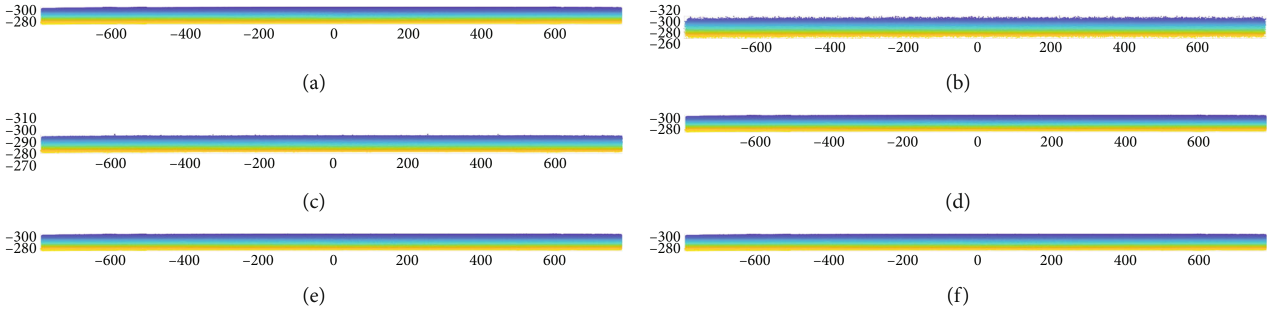
$$d = |p_i - p_m| \cdot \cos \theta \tag{6}$$

(4) Projecting $p_i \in P$ onto the local fitting plane L_i and obtaining a projection point p_i'

$$p_i' = p_i - d \cdot n_i \tag{7}$$

(5) Getting the new point set P'

2.3. Hole Filling. For the hole repair based on scattered point clouds, the scattered point cloud data can be triangulated first, and the possible holes in the point cloud data can be transformed into grid holes. When using the triangulation algorithm to fit the surface of a point cloud, the part of the triangle with a hole is usually relatively large, so in the process of repairing the point cloud hole, we can judge whether



(a)

(b)

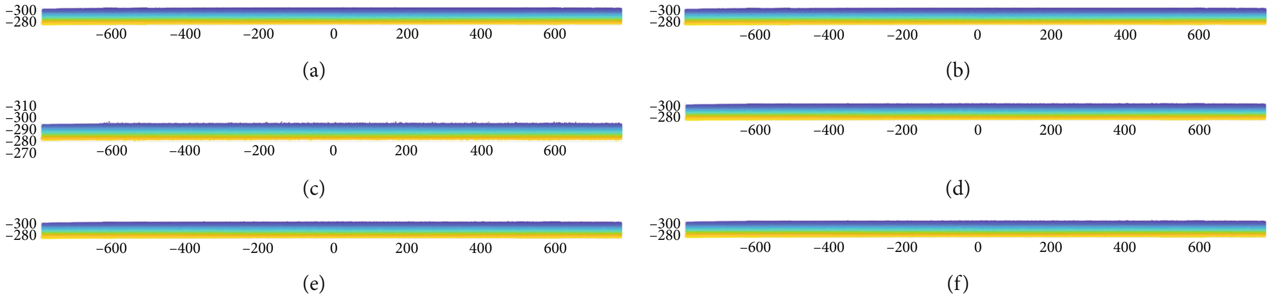
(c)

(d)

(e)

(f)

Figure 7: Denoising results of the three methods in the pantograph model. (a) The original pantograph point cloud. (b) Pantograph point cloud with 20% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 8: Denoising results of the three methods in the pantograph model. (a) The original pantograph point cloud. (b) Pantograph point cloud with 30% noise. (c) Statistical filtering algorithm. (d) Radius filtering algorithm. (e) Our outlier removal algorithm. (f) Our confounding point removal algorithm.

there is a hole by the area of the triangle. In this paper, a priority value is established preferentially to repair the part with a larger hole area in the point cloud. A higher priority value means that a specific part in the point cloud has a larger area of the hole. Then, a point is inserted firstly in the part with the largest priority value. Next, the inserted point is merged with the original point, and the above operations are repeated until the holes are repaired.

The specific implementation steps are as follows:

(1) Triangulating the target point cloud using triangulation algorithm

(2) Getting the corresponding three edges a_i, b_i, c_i from the vertices of each triangular patch; and using Heron's formula to calculate the area s_i of each triangular patch; then getting the priority value corresponding to each triangular patch

$$p_{si} = \frac{a_i + b_i + c_i}{2}, \tag{8}$$

$$s_i = \sqrt{p_{si}(p_{si} - a_i)(p_{si} - b_i)(p_{si} - c_i)} \tag{9}$$

(3) Getting the largest priority value, that is, finding the triangular patch with the largest area and its vertices

(4) Inserting a point P_{hi} at the center of gravity of the triangle with the largest priority value and getting a new point set

(5) Repeating (1) to (4) until the hole is repaired completely

3. Experimental Results and Data Analysis

3.1. Algorithm Validation. The denoising accuracy and denoising recall rate [25] are adopted to evaluate the effect of the algorithm on the outlier removal so as to verify the effectiveness of the proposed algorithm. Since the denoising accuracy and the denoising recall rate are inversely proportional, the F-score is introduced as the harmonic mean of the two parameters. A higher F-score means a better performance of the algorithm. The chamfer distance [26] is used for measuring the average nearest squared distance between the point cloud S_1 and S_2.

The denoising accuracy can be calculated as follows:

$$P_d = \frac{N_q}{N_s}, \tag{10}$$

where P_d is the denoising accuracy, N_q is the number of noisy points removed, and N_s means the whole number of noisy points.

The denoising recall rate is defined by the following formula:

$$R_d = \frac{N_q}{N_y}, \tag{11}$$

Table 5: Denoising results of the three methods with 10% noise in pantograph point cloud.

Algorithms	Total number of noisy point clouds	Pd	Rd	*F*1
Statistical filtering	695990	0.7934	0.9812	0.8774
Radius filtering	695990	0.8377	0.9318	0.8823
Our method	695990	0.8319	0.974	0.8974

Table 6: Denoising results of the three methods with 20% noise in pantograph point cloud.

Algorithms	Total number of noisy points	Pd	Rd	*F*1
Statistical filtering	759262	0.7569	0.9687	0.8498
Radius filtering	759262	0.8358	0.9666	0.8964
Our method	759262	0.8358	0.9811	0.9028

where R_d is the denoising recall and N_y is the number of points that should be removed.

So the F-score can be obtained by

$$F1 = \frac{2 * P_d * R_d}{P_d + R_d}, \tag{12}$$

where F_1 is the F-score.

We defined $d_{CD}(S_1, S_2)$ as the chamfer distance between the point cloud S_1 and S_2, and it can be calculated as follows:

$$d_{CD}(S_1, S_2) = \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2, \tag{13}$$

where x is the point in S_1 and y is the point in S_2.

In the experiments, we added 10%, 20%, and 30% random noise to the point cloud, respectively. The method of adding noise points in this paper is as follows: some points are selected randomly in the origin point cloud; then, the coordinates of these points are changed within a range; here, we used the 2 to 8 times of mesh resolution, so the confounding points and outliers could be obtained, adding these points into the origin point cloud so that we can get the noisy point cloud finally.

For enhancing the quality of the point cloud, first, we removed the outliers using the proposed algorithm and compared it with the statistical filtering and the radius filtering. As shown from Figures 1–3 and Tables 1–3, four neighborhood points are selected in the four algorithms, and the highest F-score value of each algorithm and the corresponding P_d and R_d are recorded. In the second part, we removed the confounding points using the proposed method. After processing the noisy point cloud by the proposed method, we calculated the d_{cd} in each part. Finally, the point cloud was triangulated, and a priority value was set. Points are preferentially inserted to where the priority value was the largest so as to repair the holes.

Table 7: Denoising results of the three methods with 30% noise in pantograph point cloud.

Algorithms	Total number of noisy point clouds	Pd	Rd	*F*1
Statistical filtering	822534	0.7122	0.9512	0.8145
Radius filtering	822534	0.8334	0.9787	0.9002
Our method	822534	0.8336	0.9791	0.9004

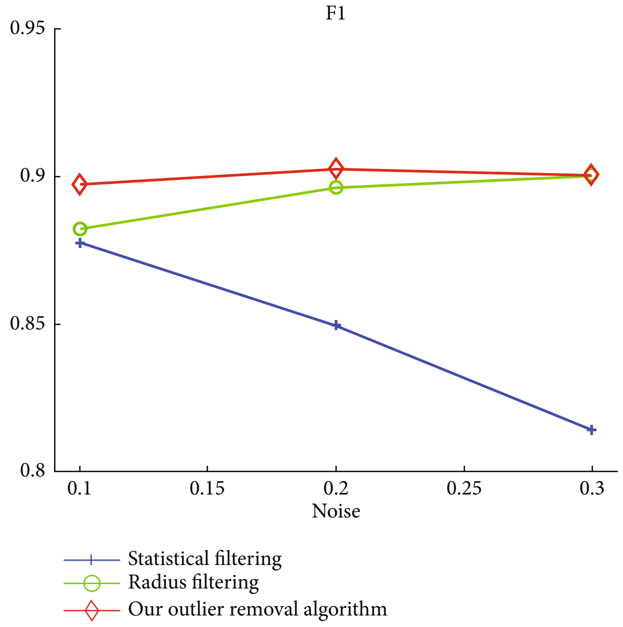


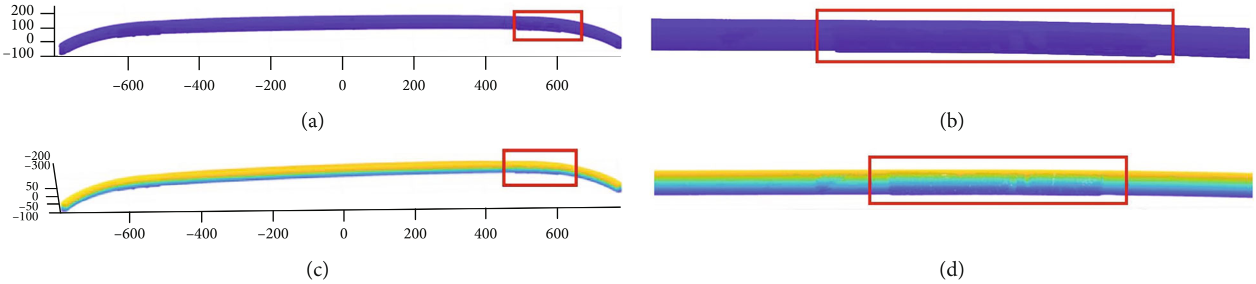
Figure 9: F-score of the three algorithms.

Table 8: Chamfer distance in pantograph point cloud.

Noise level	$d_{CD}(s_0, s_1)$	$d_{CD}(s_0, s_2)$	$d_{CD}(s_0, s_3)$
10%	2.2505	0.0181	0.0163
20%	4.1081	0.0325	0.0273
30%	5.5758	0.0532	0.0447

Figure 1 and Table 1 show that after the Bunny point cloud with 10% noise is processed by the proposed method, the surface of the point cloud is smooth, uniform, and almost without noise; meanwhile, the outliers and confounding points are removed. The denoising recall rate of the radius filtering is higher than that of the statistical filtering and the proposed outlier removal algorithm. But the denoising accuracy rate and F-score of the outlier removal algorithm are better than those of the statistical filtering and the radius filtering.

Figure 2 and Table 2 demonstrate that after the Bunny point cloud with 20% noise is processed by the proposed method, no noticeable noise points appear on the surface of the point cloud, together with the outliers and confounding points removed. The denoising recall rate of the radius filtering is better than that of the statistical filtering and the outlier removal algorithm. Yet the denoising accuracy rate



(a) (b)

(c) (d)

Figure 10: Hole filling effect. (a) Original point cloud. (b) Local hole in original point cloud. (c) Point cloud after hole-filling. (d) Local hole after hole-filling.

and F-score of our outlier removal algorithm are better than those of the statistical filter and radius filtering.

In experiment 3, Figure 3 and Table 3 show that after in the Bunny point cloud with 30% noise is processed by the proposed method, no obvious noise points exist on the surface of the point cloud, together with the outliers and confounding points removed. The denoising recall rate of the radius filtering is better than that of the statistical filtering and the proposed outlier removal algorithm. But the denoising accuracy rate and F-score of our outlier removal algorithm are better than those of the statistical filter and radius filtering.

The effect comparison of the three algorithms in removing outliers with different noise degrees is shown in Figure 4. The proposed outlier removal algorithm shows an upward trend with the increase of noise level, the radius filtering tends to be stable, and the statistical filtering shows a downward trend. The algorithm in this paper has always been better than the other two algorithms.

The s_0 is the original Bunny point cloud, s_1 is the noisy point cloud, s_2 is the point cloud after the outliers removed by the proposed outlier removal algorithm, and s_3 is the point cloud after the confounding points removed by the proposed confounding point removal algorithm. From the evaluation indicators in Table 4, we can see that $d_{CD}(s_0, s_3)$ is smaller than $d_{CD}(s_0, s_2)$ and $d_{CD}(s_0, s_1)$, indicating that the proposed algorithm can effectively remove outliers and confounding points.

The Bunny point cloud with 20% noise added is taken as an example to verify the effect of hole filling. Both the Bunny point cloud with noise and that has been processed by the proposed algorithm are triangulated, as shown in Figures 5(a) and 5(b). Figures 5(c) and 5(d) show the triangulation effect of the point cloud with noise and that has been processed by the proposed algorithm after the viewing angle is flipped, respectively. The noise on the surface of the Bunny noisy point cloud is completely removed, and the fitted surface is uniform and smooth without apparent loss. The holes at the bottom of the point cloud are also effectively filled. The results show that the proposed algorithm in this paper preserves the sharp and detailed features of the point cloud while removing the outliers and confounding points and makes an effective repair for the holes finally.

3.2. Experimental Verification of Algorithm Performance. To further verify the effect of the proposed algorithm, the actual point cloud data of the locomotive pantograph slide, obtained by the laser line structured light three-dimensional scanner, is set as the experimental object. In the data collection, the camera type is Ranger3, the laser model is Lingyun (808 nm, 15 W), and the effective accuracy of the 3D scanner is 0.5 mm. The collected pantograph data contains 632,718 points with some obvious defects on the surface. The results are shown from Figures 6–8 and Tables 5–7.

Figure 6 and Table 5 show that after the pantograph point cloud with 10% noise is processed by the proposed method, no obvious noise points exist on the surface of the point cloud, together with the outliers and confounding points removed. Radius filtering has the highest denoising accuracy. The denoising recall rate of the statistical filtering is better than that of the radius filtering and the proposed outlier removal algorithm. However, the denoising accuracy of the proposed outlier removal algorithm is close to that of the radius filtering, and the denoising recall rate is close to that of the statistical filtering. The F-score of the proposed outlier removal algorithm is better than that of the statistical filtering and radius filtering. So the comprehensive performance of the proposed method is the best.

Figure 7 and Table 6 show that after the pantograph point cloud with 20% noise is processed by the proposed method, no noticeable noise points appear on the surface of the point cloud, together with outliers and confounding points removed. The denoising accuracy rate, the denoising recall rate, and F-score of the proposed outlier removal algorithm are better than those of the statistical filtering and radius filtering.

Figure 8 and Table 7 illustrate that after the pantograph point cloud with 30% noise is processed by the proposed method, no evident noise points remain on the surface of the point cloud together with the outliers and confounding points removed. The denoising recall rate of the radius filtering is better than that of the statistical filtering and the proposed outlier removal algorithm, but the denoising accuracy rate and F-score of the proposed outlier removal algorithm are better than those of the statistical filter and radius filtering.

The effect comparison of the three algorithms in removing outliers with different noise degrees is shown in Figure 9. The proposed outlier removal algorithm and the radius filtering show an upward trend with the increase of noise level, and the statistical filtering shows a downward tread. The proposed algorithm always performs better than the other two algorithms.

The s_0 is the original pantograph point cloud, s_1 is the noisy point cloud, s_2 is the point cloud after the outliers removed by the proposed outlier removal algorithm, and s_3 is the point cloud after the confounding points removed by the proposed confounding point removal algorithm. From the evaluation indicators in Table 8, the proposed algorithm can remove both the outliers the confounding points.

To show the hole filling effects, we compared the original pantograph point cloud with that after the hole filling, as shown in Figure 10. Figures 10(a) and 10(c) illustrate the original pantograph point cloud and that after the hole filling, respectively. Figures 10(b) and 10(d) demonstrate the original pantograph point cloud with the local hole and that after the hole filling. The results show that the algorithm proposed in this paper effectively fills the hole.

4. Conclusions

In this paper, an algorithm for point cloud quality optimization and enhancement is designed and verified. Firstly, an adaptive octree is established to rasterize the point cloud and to calculate the density of each grid. Then, the statistical filtering is combined to remove the outliers from the point cloud data. And the method of plane projection is used for removing the confounding points. Finally, the point cloud is triangulated, and a priority value is set up to insert the point to the place where the priority value is the largest, so as to repair the holes. To verify the effectiveness of the proposed algorithm, we, respectively, added 10%, 20%, and 30% noise to the Bunny point cloud and used the actual point cloud data of the locomotive pantograph slider to test. Then, the proposed algorithm was compared with the statistical filtering algorithm and the radius filtering algorithm. Experimental results show that the proposed algorithm can decently keep the sharp and detailed features from the original point cloud surface while removing noise. It can also effectively fill the holes and comprehensively perform better than the algorithms of statistical filtering and radius filtering.

Data Availability

Data will be available on request.

Conflicts of Interest

The authors declare no competing financial interests.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (No. 61960206010).

References

[1] H. Qian, W. Zeyong, L. Jinlong, J. Wennan, and G. Xiaorong, “Automatic recognition of bolts on locomotive running gear based on laser scanner 3D measurement[J],” *Opto-electronic Engineering*, vol. 45, no. 1, article 170532, 2018.

[2] J. Wang, K. Xu, L. Liu et al., “Consolidation of low-quality point clouds from outdoor scenes,” *Computer Graphics Forum*, vol. 32, no. 5, pp. 207–C216, 2013.

[3] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, “Edge-aware point set resampling,” *ACM Transactions on Graphics*, vol. 32, no. 1, pp. 1–12, 2013.

[4] X. Zhan, Y. Cai, and P. He, “A three-dimensional point cloud registration based on entropy and particle swarm optimization,” *Advances in Mechanical Engineering*, vol. 10, no. 12, 2018.

[5] M. Li and C. Sun, “Refinement of LiDAR point clouds using a super voxel based approach,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 143, no. 9, pp. 213–221, 2018.

[6] F. Pirotti, R. Ravanelli, F. Fissore, and A. Masiero, “Implementation and assessment of two density-based outlier detection methods over large spatial point clouds,” *Open geospatial data, softw. stand.*, vol. 3, no. 1, 2018.

[7] M. Zeybek and İ. Şanlıoğlu, “Point cloud filtering on UAV based point cloud,” *Measurement*, vol. 133, pp. 99–111, 2019.

[8] M. Zeybek, “Inlier point preservation in outlier points removed from the ALS point cloud,” *Beiträge zur experimentellen Pathologie und Physiologie*, vol. 49, no. 10, pp. 2347–2363, 2021.

[9] S. Fleishman, D. Cohen-Or, and C. T. Silva, “Robust moving least-squares fitting with sharp features,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 544–552, 2005.

[10] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, “Parameterization-free projection for geometry reconstruction[C],” *Proceedings of International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 2007)*, vol. 22, 2007.

[11] A. Huang, D. Li, A. Zhang, U. Ascher, and D. Cohen-Or, “Consolidation of unorganized point clouds for surface reconstruction,” *ACM series on computing methodologies.*, vol. 28, no. 5, pp. 1–7, 2009.

[12] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D Point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[13] X. Ning, F. Li, and G. Tian, “An efficient outlier removal method for scattered point cloud data,” *PLoS One*, vol. 13, no. 8, p. e0201280, 2018.

[14] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise[C],” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96).*, pp. 226–231, AAAI Press.

[15] E. Leal, G. Sanchez-Torres, and J. W. Branch, “Sparse regularization-based approach for point cloud denoising and sharp features enhancement,” *Sensors*, vol. 20, no. 11, pp. 3206–3218, 2020.

[16] C. Feng, J. Liang, M. Ren et al., “A fast hole-filling method for triangular mesh in additive repair,” *Applied Sciences-Basel*, vol. 10, no. 3, p. 969, 2020.

[17] S. Fleishman, I. Drori, and D. Cohen-Or, “Bilateral mesh denoising,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 950–953, 2003.

[18] “Approximate Delaunay mesh reconstruction and quality estimation from point samples,” *Journal of Computational and Applied Mathematics*, vol. 274, pp. 23–34, 2015.

[19] G. Harary, A. Tal, and E. Grinspun, “Context-based coherent surface completion,” *ACM Trans. Graph. (TOG)*, vol. 33, no. 1, pp. 1–12, 2014.

[20] L. Tao, G. Yuan, Q. Pinle, and L. Lifang, "Improved and optimized Poisson reconstruction algorithm based on point cloud library[J]," *Application Research of Computers*, vol. 35, no. 3, pp. 940–943, 2018.

[21] X. Wang, J. Hu, D. Zhang, L. Guo, H. Qin, and A. Hao, "Multiscale geometry detail recovery on surfaces via empirical mode decomposition," *Computers and Graphics*, vol. 70, pp. 118–127, 2018.

[22] K. F. Leong, C. K. Chua, and Y. M. Ng, "A study of stereolithography file errors and repair. Part 1. Generic solution," *International Journal of Advanced Manufacturing Technology*, vol. 12, no. 6, pp. 407–414, 1996.

[23] L. I. U. Zhenghong and L. I. N. Yun, "A hole repairing algorithm in triangular mesh[J]," *Journal of Guiyang University Natural Sciences*, vol. 13, no. 3, pp. 101–105, 2018.

[24] L. U. Dongdong and Z. O. U. Jingui, "Comparative research on denoising algorithms of 3D laser point cloud[J]," *Bulletin of Surveying and Mapping*, vol. 52, 2019.

[25] S. Zhou, X. Liu, C. Wang, and B. Yang, "Non-iterative denoising algorithm based on a dual threshold for a 3D point cloud," *Optics and Lasers in Engineering*, vol. 126, article 105921, 2020.

[26] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: two new techniques for image matching[C]," *Proceedings of 5th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 659–663, 1977.