

Article

A Color- and Geometric-Feature-Based Approach for Denoising Three-Dimensional Cultural Relic Point Clouds

Hongjuan Gao^{1,2,3,*}, Hui Wang¹ and Shijie Zhao¹

¹ School of Information Engineering, Ningxia University, Yinchuan 750021, China; 12022131976@nxu.edu.cn (H.W.); 12022132009@nxu.edu.cn (S.Z.)

² Ningxia Key Laboratory of Artificial Intelligence and Information Security for Channeling Computing Resources from the East to the West, Yinchuan 750021, China

³ Collaborative Innovation Center for Ningxia Big Data and Artificial Intelligence Co-Founded by Ningxia Municipality and Ministry of Education, Yinchuan 750021, China

* Correspondence: ghj_y2004@nxu.edu.cn

Abstract: In the acquisition process of 3D cultural relics, it is common to encounter noise. To facilitate the generation of high-quality 3D models, we propose an approach based on graph signal processing that combines color and geometric features to denoise the point cloud. We divide the 3D point cloud into patches based on self-similarity theory and create an appropriate underlying graph with a Markov property. The features of the vertices in the graph are represented using 3D coordinates, normal vectors, and color. We formulate the point cloud denoising problem as a maximum a posteriori (MAP) estimation problem and use a graph Laplacian regularization (GLR) prior to identifying the most probable noise-free point cloud. In the denoising process, we moderately simplify the 3D point to reduce the running time of the denoising algorithm. The experimental results demonstrate that our proposed approach outperforms five competing methods in both subjective and objective assessments. It requires fewer iterations and exhibits strong robustness, effectively removing noise from the surface of cultural relic point clouds while preserving fine-scale 3D features such as texture and ornamentation. This results in more realistic 3D representations of cultural relics.

Keywords: 3D point cloud; cultural relics; denoising; graph signal processing; Markov graph model; prior probability distribution; maximum a posteriori (MAP) estimation



Citation: Gao, H.; Wang, H.; Zhao, S. A Color- and Geometric-Feature-Based Approach for Denoising Three-Dimensional Cultural Relic Point Clouds. *Entropy* **2024**, *26*, 319. <https://doi.org/10.3390/e26040319>

Academic Editor: Amelia Carolina Sparavigna

Received: 14 February 2024

Revised: 31 March 2024

Accepted: 3 April 2024

Published: 5 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Three-dimensional laser scanning technology has become increasingly popular in various fields of society, such as digitization, virtual display, and virtual restoration of cultural relics. However, the acquisition process of cultural relic point clouds often results in noise in geometry and color due to the inherent limitations of 3D laser scanners or depth cameras. This noise can be caused by occlusion resulting from various view angles, reflective materials, dust on the surface of objects, light intensities, and the operation of scanning personnel [1]. The cultural relic point cloud surface typically contains significant fine details, such as ornamentation or textures, which can be intertwined with surface noise. Effectively removing noise from the surface of the cultural relic point cloud while preserving the fine-scale 3D features is a significant challenge.

In order to acquire a high-precision 3D model of a cultural relic with realistic texture, it is essential to remove noise from the raw 3D point cloud. The noise in the point cloud can be divided into two categories based on their distribution: surface noise and outliers [2]. Each outlier will usually be far away from the surface of the point cloud with a sparse neighborhood, which means that they are easy to remove using methods such as the boxplot method or special software. However, eliminating surface noise presents a greater challenge as it is often closely intertwined with the underlying surface of the 3D point

cloud. This is especially true when the surface of the 3D point cloud features texture and ornamentation.

To obtain clean point clouds for further processing, various surface smoothing techniques have been developed in the past two decades. These techniques include filtering-based methods [3–6], moving least squares (MLS)-based methods [7,8], locally optimal projection (LOP)-based methods [9,10], non-local methods [11–13], and sparsity-based methods [14–17]. Although these methods have been successful in achieving excellent denoising effects for 3D models with smooth surfaces, they have not yielded satisfactory results for point clouds of cultural relics. This often results in over-smoothing and the loss of surface details. Striking a balance between preserving fine details and achieving effective denoising with these methods is challenging.

In recent years, several methods have been proposed for denoising point clouds, including the graph feature learning method [18–26] and the deep learning method [27–33].

The effectiveness of deep learning in denoising point clouds depends heavily on factors such as the geometric structure, the scale of the data, and the noise characteristics of the training set. When faced with an unknown scene or limited data, the method based on deep learning may not necessarily outperform traditional methods. For instance, a model trained using commonly available 3D point clouds may experience a significant decrease in performance when applied to point clouds of cultural relics, which are considered to be rare samples.

Graph-based denoising methods utilize graph filters to remove noise from point clouds represented by graphs [34]. Previous methods such as graph Laplacian regularization (GLR) [20] and the feature graph learning [23] algorithm have shown promising results in inferring the underlying graph structure of clean point clouds. However, these methods primarily rely on geometric priors, making it challenging to achieve effective denoising while preserving fine detail.

We raise an interesting question: if color perception information is added to guide the graph signal processing, can a balance between denoising effectiveness and detail preservation be achieved?

To investigate this, we propose a novel 3D point cloud denoising method based on graph signal processing specifically designed for cultural relic point clouds. Our contributions are twofold. First, we incorporate not only geometric information such as 3D coordinates and surface normals but also color distribution as a feature. The use of a multi-modal representation for vertex features leads to superior denoising performance. Second, we introduce a 3D point cloud simplification module to dynamically adjust the number of 3D point clouds to reduce the running time of the denoising algorithm.

This paper is organized as follows: In Section 2, we introduce previous point cloud denoising methods. In Section 3, we describe the basic concepts of graph signal processing. In Section 4, we provide the details of our proposed method, which mainly focuses on surface noise removal. In Section 5, we present the experimental results and discussion. Finally, we present our conclusions.

2. Related Work

Point cloud denoising techniques can be divided into two main types: outlier removal techniques and surface noise smoothing techniques. Outlier removal is a relatively straightforward process, as outliers are usually distinct from other data points and can be easily identified and removed. On the other hand, surface noise removal can be more challenging, as surface noise is often random and irregular and requires more sophisticated techniques to be removed. In this paper, we will primarily focus on surface noise removal methods.

Filtering-based methods: Filtering-based methods were initially used for 2D image smoothing and were later extended to denoise 3D point clouds [2]. These methods assume that the noise on the surface of point clouds is high-frequency noise, and they use filters that target vertices or face normals. Early approaches utilized Laplacian smoothing or improved Laplacian transform based on vertex positions to denoise triangular meshes. However, this

often resulted in the excessive smoothing of surface features and was not effective when dealing with large amounts of noise. In recent years, filtering-based methods have been significantly improved. Notable examples include guided normal filtering [3,4] and rolling guidance normal filtering [5], which have demonstrated successful denoising effects in practical applications [6]. Nevertheless, a major drawback of these methods is that the normal filtering process tends to blur the small-scale features of the 3D model surface, resulting in the over-smoothing of 3D models with intricate surface details.

MLS-based and LOP-based methods: Early in the development of denoising technology, moving least squares (MLS) and local optimal projection (LOP) methods were well-known and popular denoising methods. However, their denoising effect is limited, and they are no longer the mainstream methods. MLS-based methods [7] approximate the point cloud using a smooth surface and project the points from the input point cloud onto the fitted surface. These methods are unstable in cases of a low sampling rate or high curvature and are highly sensitive to outliers [8]. LOP-based methods [9] aim to find the best possible solution to represent the underlying surface within a local region of the search space while ensuring an even distribution across the input point cloud. However, these methods can suffer from over-smoothing [10].

Non-local methods: Non-local methods [11–13] establish self-similarity among surface patches in the point cloud by solving an optimization problem. However, these methods often suffer from high computational complexity when searching for non-local similar patches.

Sparsity-based methods: Sparsity-based methods [14–16] transform the denoising problem of a 3D point cloud into an optimization problem. This is achieved by obtaining a sparse representation of the surface normal by minimizing the number of non-zero coefficients with sparsity regularization. To preserve the sharp features of the 3D point cloud, either the L_0 or L_1 norm is used. It should be noted that sparsity-based methods tend to give better denoising results when the noise is small. However, for high noise levels, these methods can suffer from either over-sharpening or over-smoothing [17].

Graph-based methods: Graph-based denoising methods [18–26] transform the problem of removing noise into a graph-constrained optimization problem and perform noise removal through the structure and connectivity of the graph. However, a drawback of these approaches is that they often misestimate the local surface by relying solely on the geometry information of the vertices. In addition, the performance of graph-based denoising methods remains unstable for highly noisy point clouds.

Deep learning methods: Deep learning denoising methods [27–35] train an end-to-end neural network to remove noise. During the training stage, the model learns the mapping between noisy points and clean clouds. In the testing stage, the trained model is used to denoise point clouds with similar noise characteristics and geometric shapes. Deep learning methods are more effective at denoising and preserving fine features. However, these methods require a large volume of training data, making them time-consuming and impractical for unknown scenes. In addition, optimizing and improving the efficiency of the algorithm is also an important consideration.

Several alternative denoising methods have been proposed by other scholars. For instance, there is a point cloud denoising algorithm based on a method library [36], as well as a laser point cloud denoising method that uses principal component analysis (PCA) and surface fitting [37]. However, these methods often encounter the common issue of inadequate denoising of sharp edges, resulting in excessive smoothing.

Recently, some scholars have proposed deep-unfolding denoising [38–40] and quantum-based denoising [41,42], which have achieved competitive results compared to state-of-the-art image denoising tasks. How to draw on the ideas of these methods to denoise the point cloud is a very valuable research work in the future.

3. Preliminaries

3.1. Graph Signal and Graph Laplacian

In this section, we present a brief overview of fundamental concepts in graph signal processing. We define an undirected weighted graph for a vertex set of cardinality $|V| = N$, where the edge set E connects vertices of the form $(v_i, v_j) \in V$. Each edge is assigned a non-negative weight $w_{i,j}$, and the adjacency matrix W is a $N \times N$ real matrix with values ranging from 0 to 1. The combinatorial graph Laplacian is defined as $L := D - W$, where D represents the degree matrix of the graph G , with $d_{ij} = \sum_{j=1}^N w_{ij}$ denoting the degree of each vertex.

3.2. Graph Laplacian Prior

Graph signal data reside on the vertices of a graph, which include 3D coordinates, normal vectors, and color information on a 3D point cloud. A graph signal z is considered to be smooth with respect to the topology of G if it satisfies the following conditions:

$$z^T L z = \sum_{i=1}^N \sum_{j=1}^N w_{i,j} (z_i - z_j)^2 < \epsilon, \quad (1)$$

where ϵ is a positive scalar, and the Laplacian matrix L is a symmetric positive semi-definite matrix. The larger w_{ij} is, the more similar z_i and z_j are and the smaller the value of $z^T L z$ is.

Formula (1) forces signal z to adapt to the topology of G , which is referred to as graph Laplacian regularization (GLR), also known as graph signal smoothness prior. By minimizing the graph Laplacian regularization term, the signal can be smoothed. This prior is used in our paper to remove the surface noise, as discussed in Section 4.

If we consider $L(z)$ as a function for signals z , then reweighted prior is redefined as

$$z^T L(z) z = \sum_{i=1}^N \sum_{j=1}^N w_{i,j}(z_i, z_j) \cdot (z_i - z_j)^2, \quad (2)$$

where $w_{i,j}(z_i, z_j) = \exp\left\{-\frac{(z_i - z_j)^2}{\sigma^2}\right\}$, and $w_{i,j}(z_i, z_j)$ is the (i, j) -th element of the corresponding adjacency matrix W .

4. The Proposed Method

Considering a point cloud contaminated by noise Y' with a Gaussian noise distribution, our basic strategy can be viewed from the theory of graph signal processing; our goal is to move noisy points to the underlying surface to generate a clean point cloud Y .

Figure 1 illustrates the implementation of our denoising approach, which consists of the following four modules:

- Simplification of 3D point cloud: reducing the collection of points to a smaller subset that retains its fundamental topology, thereby reducing the running time of the denoising algorithm;
- Definition of graph signals: interpreting the geometric and color information of the vertices in the input point clouds as graph signals; the geometric information includes geometry coordinates and normal vectors;
- Graph construction and feature graph learning: defining local patches within the point cloud and constructing a graph model with Markovian properties; using a feature graph learning scheme to determine edge weights and solving a maximum a posteriori (MAP) estimation problem with GLR as the signal prior;
- Application of an optimization algorithm to enforce smoothness on the graph signal: we alternately optimize the feature metric matrix M by minimizing the GLR, and M and noisy point cloud are updated alternately until the algorithm converges, and finally, we obtain the clean point cloud.

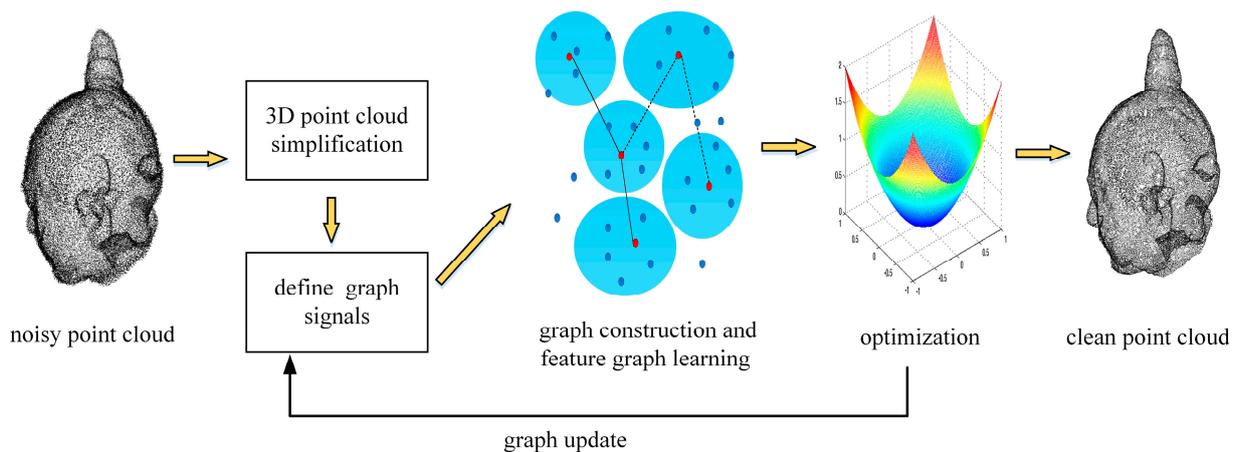


Figure 1. The proposed denoising approach.

4.1. Three-Dimensional Point Cloud Simplification

High-precision 3D artifact point clouds usually have many points, inevitably leading to high computational complexity and long processing times during denoising. Therefore, it is necessary to simplify the raw data to a more appropriate size without affecting the denoising effect. We use the method described in [43] for simplifying point clouds. The simplification process is divided into the following four steps:

- A bounding box for the point cloud is created. A local kd–tree consisting of 27 cubes of size $3 \times 3 \times 3$ is constructed. The advantage of using this form to organize the points in the point cloud is that the neighborhood points and leaf nodes of the given point can be accurately identified.
- Five feature indexes are calculated to extract features from the point cloud. These five feature indexes include the curvature feature index, the density feature index, the edge feature index, the terrain feature index, and the 3D feature index, denoted as a , b , c , d , and e , respectively. The advantage of this multiple-feature indexing approach is that it can deal with different types of point clouds and discover more intrinsic characteristics of the point cloud.
- The weights of the five feature indexes are calculated using the analytic hierarchy process (AHP) method based on data features. Assuming that w_i is the weight index of feature indexes a , b , c , d , and e , the quantification result of point p is $z_p = [a_p, b_p, c_p, d_p, e_p] [\omega_a, \omega_b, \omega_c, \omega_d, \omega_e]^T$.
- Points with larger z -values are identified as feature points, and points with smaller z -values are identified as non-feature points. All feature points form a simplified point cloud. According to the kd–tree constructed earlier, if there is no feature point in each leaf node, the non-feature point closest to the center of gravity of the node is selected to be added to the simplified point cloud.

4.2. Defining the Graph Signal by Combining Geometry and Color

Color, an important piece of information of a point cloud, has been used for 3D model retrieval [44] and point cloud segmentation [45,46]. The combination of color and geometry can positively affect graph construction, which is more semantically meaningful than using geometry alone [47]. In this paper, we aim to use both the color and geometry attributes of a vertex in the point cloud to investigate their crucial role in denoising. Figure 2 illustrates the color and geometry information of the point cloud.

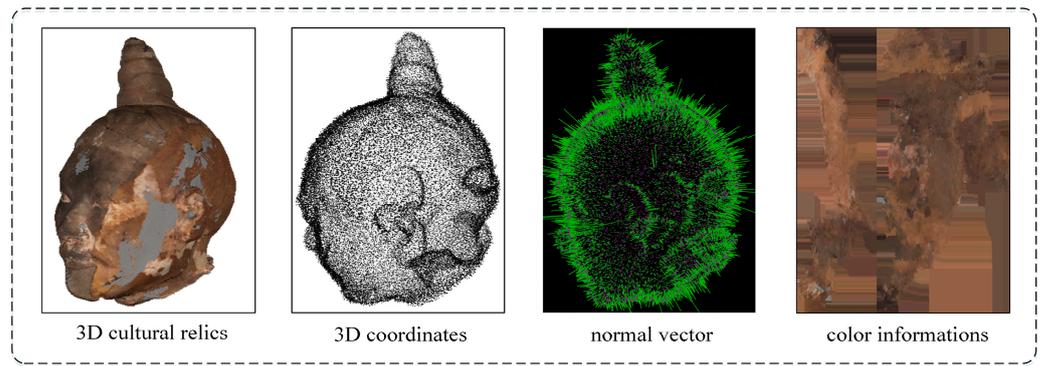


Figure 2. Geometry and color information of a vertex in a point cloud.

We constructed a k -NN graph with Markovian properties, where each vertex is connected to its k -nearest neighbors by connecting edges with associated weights. In addition to using the 3D coordinates and normal vector of the vertices as signals, we added the color attributes as graph signals. The feature vector of a vertex in the graph is denoted as

$$s_i = [P_i \ N_i \ C_i] \in R^9, \tag{3}$$

where 3D coordinates $P_i = [x_i, y_i, z_i] \in R^3$, normal vector $N_i = [n_x^i, n_y^i, n_z^i] \in R^3$, and RGB color information $C_i = [c_r^i, c_g^i, c_b^i] \in R^3$ for vertex v_i .

The normal vector is one of the important properties of the points in a point cloud. The normal vector of a point cloud is the orientation of each point in a point cloud. For example, the direction of the normal vector in Figure 3 points to the outside of the surface of the point cloud. The normal vector of a point cloud is usually a 3D vector that describes the normal properties of the point cloud surface, such as the flatness, curvature, and normal variation in the point cloud surface.

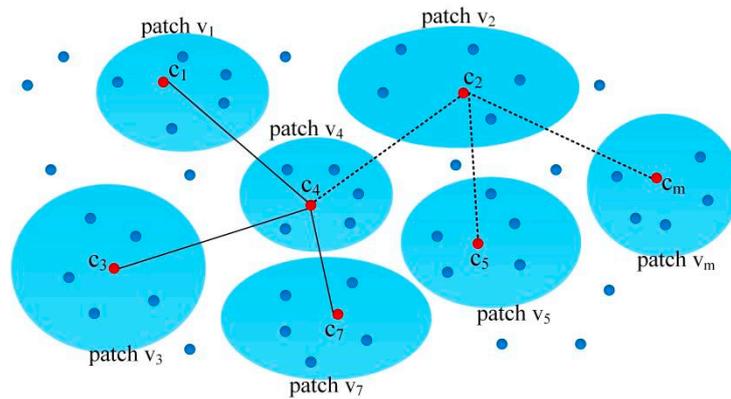


Figure 3. Illustration of the patch v_2 , v_7 and their adjacent patches.

4.3. Constructing a Graph Based on Self-Similarity Theory

We followed the method described in [26] to define the local patches in a point cloud. These patches may overlap with each other. We assumed that these patches were self-similar [48] and established connections between corresponding points, forming a k -NN graph. We considered each local patch of the point cloud as a matrix, which had a low rank. Consequently, the problem of denoising the point cloud can be reformulated as a task for minimizing the rank of the matrix.

In this study, we used a uniform sampling method to select m center points $c_i \in R^3$ from point cloud Y . For each center point c_i , we used the k -nearest-neighbor (k -NN) [49] algorithm to identify k -nearest-neighbor points in terms of Euclidean distance. A patch

v_i is a set of points that is composed of one center point c_i and k nearest neighbors. The number of nearest neighbors m was determined based on an empirical value, denoted as $m \leq N, (k + 1)m \geq N$, where N is the number of points in the point cloud Y . As a result, we obtained m local patches from the point cloud Y , as shown in Figure 3. The collection of all points within these patches is referred to as a patch set, denoted as $V \in \mathbb{R}^{(k+1)m \times 3}$.

Then, we identified ε adjacent patches for patch v_i . We used the k -NN algorithm to find the ε nearest center point for the center point c_i of patch v_i in a set of patches. The ε patches that the ε nearest center points are located in are recognized as adjacent patches of v_i . As shown in Figure 3, $\varepsilon = 3$, and the adjacent patches of v_2 are v_4, v_5 , and v_m . For a point $p_i \in v_s$, there exists a nearest corresponding point $p_j \in v_t$, and the Euclidean distance between p_i and p_j is the smallest.

In the process of constructing the local patches mentioned above, each patch is only related to its adjacent patches. The vertices in the patch are only connected to the corresponding vertices in the adjacent patch. As a result, these vertices and edges form a graph model with Markov properties.

4.4. Graph Feature Learning

We aim to calculate an optimal Mahalanobis distance $\delta_{i,j}$ for the given signals, which are represented as length-9 vectors of relevant features in a graph. We assumed two sets, $\mathcal{O}_k(i)$ and $\mathcal{O}_k(j)$, which denote the k -nearest neighbors to vertices v_i and v_j , respectively. If $p_j \in \mathcal{O}_k(i)$ or $p_i \in \mathcal{O}_k(j)$, then

$$\delta_{i,j} = -\frac{\|P_i - P_j\|^2}{\theta_P^2} - \frac{\|N_i - N_j\|^2}{\theta_N^2} - \frac{\|C_i - C_j\|^2}{\theta_C^2}, \tag{4}$$

where N_i and N_j represent the normal vector of vertices v_i and v_j ; C_i and C_j represent the color information of vertices v_i and v_j ; θ_P and θ_N represent the relative contribution of the 3D coordinates and normal vectors in the constructed graph; and θ_C represents the relative contribution of color.

Defining $s_i = [p_i, N_i, C_i]^T$, we express (4) in matrix form as

$$\delta_{i,j} = (s_i - s_j)^T \begin{bmatrix} 1/\theta_P^2 & 0 & 0 \\ 0 & 1/\theta_N^2 & 0 \\ 0 & 0 & 1/\theta_C^2 \end{bmatrix} (s_i - s_j), \tag{5}$$

where $s_i - s_j$ is the feature difference between the two connected nodes p_i and p_j . The appropriate parameters θ_P, θ_N , and θ_C play an important role in achieving good denoising performance. How to determine these parameters is the next aspect to consider.

The 3D coordinates, normal vector, and color information are features of different scales. In this context, we used the Mahalanobis distance as a measure of the similarity between the two signals. The Mahalanobis distance δ_{ij} is written as

$$\delta_{i,j} = (s_i - s_j)^T M (s_i - s_j), \tag{6}$$

where $M \in \mathbb{R}^{k \times k}$ is the Mahalanobis distance matrix, which is a measure of the relative importance of individual features in the calculation of δ_{ij} .

In the context of a graph, the edge weight $w_{i,j}(s_i, s_j)$ represents the similarity of the signals between two samples. We define the edge weight $w_{i,j}(s_i, s_j)$ using the Gaussian kernel, a commonly used method, which guarantees that the resulting graph Laplacian matrix L is positive semi-definite.

$$w_{i,j}(s_i, s_j) = \exp\{-\delta_{ij}\}, \tag{7}$$

The GLR, expressed in Formula (2), is redefined as

$$s^T L(M) s = \sum_{i=1}^N \sum_{j=1}^N \exp\left\{-\left(s_i - s_j\right)^T M \left(s_i - s_j\right)\right\} \cdot \left(s_i - s_j\right)^2. \tag{8}$$

4.5. Optimization Algorithm

We considered the solution of a clean point cloud as a feature graph learning problem. As discussed in [23], we minimized the GLR and determined the appropriate underlying graph based on signal z .

Additionally, we assumed a point cloud with added noise, namely

$$Y' = Y + E, \tag{9}$$

where $Y' \in R^{N \times 3}$ denotes the 3D coordinates of the point cloud with added noise, $Y \in R^{N \times 3}$ denotes the 3D coordinates of the clean point cloud, and $E \in R^{N \times 3}$ denotes the white Gaussian noise (AWGN) [50] that appears near the underlying surface. The AWGN has zero mean and standard deviation.

$$E \sim \mathcal{N}\left(0, \sigma^2 I\right). \tag{10}$$

Given a noisy set V' , the goal is to minimize the noise and obtain a noiseless set V . This is achieved by applying the maximum a posteriori criterion, which involves finding the most probable V given the observed V' .

$$\tilde{V}_{\text{MAP}}\left(V'\right) = \operatorname{argmax}_Y P\left(V'|V\right) P\left(V\right), \tag{11}$$

where $P(V)$ is the prior probability distribution of V , and $P(V'|V)$ is the likelihood function.

In the case of additive Gaussian white noise, the likelihood function is defined as follows:

$$P\left(V'|V\right) = P\left(Y'|Y\right) = \exp\left\{\left(-1/\left(2\sigma^2\right)\right)\|Y - Y'\|_F^2\right\}, \tag{12}$$

where $\|\cdot\|_F^2$ is the Frobenius norm.

If G is a graph with Markov properties [51], and GLR is taken as the prior probability distribution of the set V , then the following is evident:

$$P\left(V\right) = \exp\left\{-\beta \operatorname{tr}\left(V^T L\left(M\right) V\right)\right\}, \tag{13}$$

where $\beta = \left(2\pi\right)^{-\frac{n-1}{2}}\left(\left|L\left(M\right)\right|^*\right)^{\frac{1}{2}}$ and M is the Markov distance matrix.

The denoising formula can be obtained by combining (11)–(13).

$$\begin{aligned} \min_{Y, M} & \|Y - Y'\|_F^2 + \left(2\sigma^2\beta\right)\operatorname{tr}\left(V^T L\left(M\right) V\right), \\ \text{s.t. } & M \succ 0; \operatorname{tr}\left(M\right) \leq C^*. \end{aligned} \tag{14}$$

It should be noted that C^* is a constraint parameter closely related to the algorithm performance.

Denoising a 3D point cloud is an iterative process. In the first iteration, M is initialized with the identity matrix. Then, the Laplacian matrix $L(M)$ is computed, and the conjugate gradient method [52] is used to solve it. In the subsequent iteration, M is updated, and the optimization problem of M is solved using the near-end gradient method (PG) [53]. The values of M and Y are updated alternately until they converge.

The optimization algorithm is presented in Algorithm 1.

Algorithm 1: Optimization algorithm

Input: Noisy point cloud Y' , number of patches m , number of nearest neighbors k , number of adjacent patches ε , trace constraint C^* .
Output: Denoised point cloud Y .

- 1 Initialize Y with Y' ;
- 2 for iter = 1, 2, ... do
- 3 estimate normal for Y ;
- 4 initialize m empty patches V ;
- 5 find the adjacent ε patches;
- 6 initialize M with identity matrix;
- 7 compute the feature distance $s_i - s_j$ for each vertex pair (i, j) ;
- 8 solve M ;
- 9 compute adjacency matrix W over all patches;
- 10 compute Laplacian matrix L ;
- 11 solve Y with (14);
- 12 end

5. Experiment Results and Analysis**5.1. Experiment Environment and Dataset**

Our method was implemented on a desktop computer running MS Windows 10. The computer was equipped with an Intel® Core™ i9-9900k CPU (3.60 GHz), 64 GB of RAM, and two GeForce RTX 2070 GPUs. We used MATLAB R2019b programming for the implementation.

To demonstrate the state of the art of our approach, we performed experiments on 3D point clouds of terracotta warrior fragments, tiles from the Qin Dynasty, and Tang tri-color Hu terracotta sculptures, as shown in Figure 4. We achieved the best performance of the algorithm by selecting the optimal parameters. We repeated the experiments thirty times and calculated the average results for three metrics: *SNR*, *MSE*, and running time.



Figure 4. Three-dimensional point clouds of cultural relics: (a–f) terracotta warrior fragments numbered G3-I-b-70, 4#yt, G10-52, G10-46-5, G3-I-C-94, and G10-11-43(47); (g,h) Qin Dynasty tiles numbered Q002789 and Q003418; (i,j) Tang tri-color Hu terracotta sculptures numbered H73 and H80.

5.2. Evaluation Metrics

The evaluation of the denoising results was performed using visual effects, *SNR*, and *MSE*, following recent point cloud denoising research. Let us assume that the real point cloud and the predicted point cloud are denoted as $U = \{u^i\}_{i=1}^{N_1}$ and $V = \{v^i\}_{i=1}^{N_2}$, respectively. $u^i, v^i \in R^3$, and N_1 and N_2 may not be equal here.

To measure the fidelity of the denoising result, we used *MSE*, which is the minimum absolute error sum of the normal direction difference between the noisy point cloud and the denoised point cloud. A lower *MSE* value indicates a better denoising effect. The calculation of the *MSE* is as follows:

$$MSE = \frac{1}{2N_1} \sum_{u_i \in U} \min_{v_j \in V} \|u_i - v_j\|_2^2 + \frac{1}{2N_2} \sum_{v_i \in V} \min_{u_j \in U} \|v_i - u_j\|_2^2 \quad (15)$$

The *SNR* is a measure of the signal-to-noise ratio in a 3D point cloud, usually expressed in decibels. A higher signal-to-noise ratio indicates better denoising reliability of the algorithm. The *SNR* can be calculated using the following formula:

$$SNR = 10 \log \frac{1/N_2 \sum_{v_i \in V} \|v_i\|_2^2}{MSE} \quad (16)$$

5.3. Algorithm Performance Analysis

5.3.1. The Effect of Parameters on Algorithm Performance

Our algorithm has four main parameters: the number of patches *m*, the number of points in each patch *k*, the number of nearest neighboring patches *ε*, and the constraint parameter *C**. Among these parameters, *C** has a significant impact on the denoising effect. Therefore, it is important to determine the optimal value for *C*. To do so, we can first choose an initial value based on experience and then explore values around this initial value with a certain step size to determine the optimal value.

In this study, we analyzed the effect of parameter values on the *MSE* and *SNR* under Gaussian white noise with standard deviations $\sigma = 0.02$, $\sigma = 0.05$, and $\sigma = 0.1$. To illustrate this, we chose the 3D fragment numbered G3-I-b-70 as our experimental data source.

As illustrated in Figure 5, the blue line represents the *MSE* value for noise with a σ of 0.02. The red trend line represents the *MSE* value for noise with a σ of 0.05, while the green trend line represents the *MSE* value for noise with a σ of 0.1. When the value of *C** is 0, the minimum *MSE* value on the blue line is 0.442, indicating the algorithm’s optimal denoising effect of the algorithm at this point. Similarly, when the value of *C** is 0.1, the minimum *MSE* on the red line is 0.795, signifying the best denoising effect. Lastly, with a *C** value of 0.3, the minimum *MSE* on the green line is 0.899, denoting the optimal denoising effect of the algorithm at this particular point.

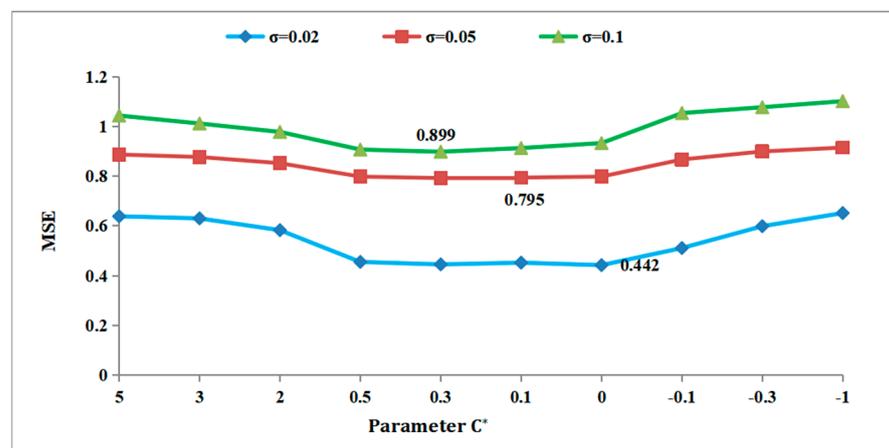


Figure 5. Effect of *C** on *MSE* for G3–Ib–70.

As shown in Figure 6, as the noise levels vary, and the *C** value changes, the denoising effect of the algorithm, as indicated by the *SNR* value, remains consistent with the *MSE* value. When the value of *C** is 0.3, the maximum *SNR* on the green line is 57.884, indicating the best denoising effect. Similarly, when the value of *C** is 0.1, the maximum *SNR* value

on the red line is 59.119, indicating the optimal denoising effect of the algorithm at this point. Finally, with a C^* value of 0, the maximum SNR on the blue line is 64.998, indicating the optimal denoising effect of the algorithm at this particular point.

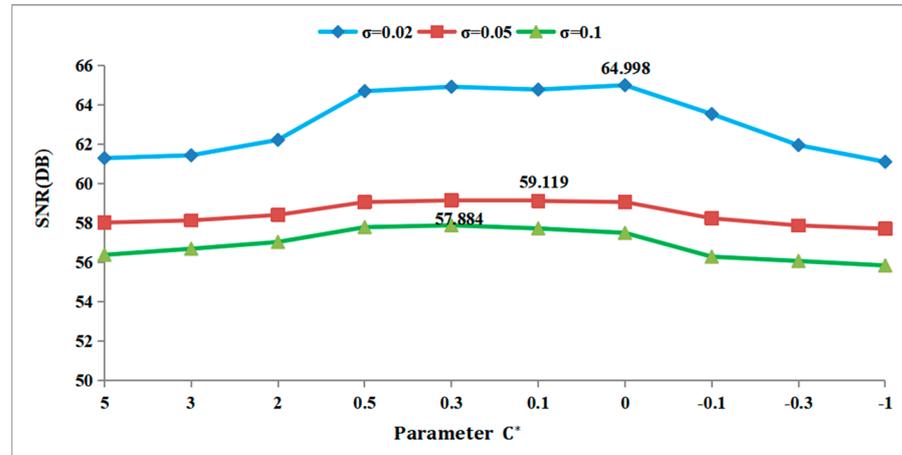


Figure 6. Effect of C^* on SNR for G3–I–b–70.

5.3.2. Ablation Experiment

In this study, we present the experimental results for two proposed methods: one using only geometry and the other using both geometry and color. We identified four main parameters that gave the best results for the proposed approach, as shown in Table 1.

Table 1. Parameter setting.

Parameter	Down Sampling Rate	The Number of Points in the Patches	The Number of Nearest Neighbors of a Patch	C^*
Value	0.3	9	10	3

The algorithm was evaluated based on four quantitative indicators: SNR, MSE, iterations, and running time. For the sake of clarity, the subjective results of the comparison between the proposed algorithm using only geometry and combined geometry and color are presented in Table 2. To illustrate this, we chose the 3D fragment numbered 4#yt as our experimental data source.

Table 2. Comparison of the proposed approach using only geometry and combined geometry and color on 4#yt.

Method	σ	Points	SNR (DB)	MSE	Iterations	Running Time (s)
Only geometry	0.02	58,380	60.66	0.61	4	172
Geometry + color			63.98	0.50	4	171
Only geometry	0.05	58,380	61.91	0.84	10	864
Geometry + color			62.14	0.72	11	870
Only geometry	0.1	58,380	59.50	1.07	46	3985
Geometry + color			62.03	0.81	49	4011
Only geometry	0.2	58,380	52.98	2.05	100	8884
Geometry + color			54.02	1.90	106	9003

Table 2 shows that the SNR and MSE of the denoising algorithm with the combined geometry and color serving as graph signals outperform the geometry-only approach. Furthermore, the inclusion of color information does not result in a significant increase in iterations or running time.

As shown in Table 3, the 3D point cloud is simplified by reducing the number of points from 58,380 to 30,000. When $\sigma = 0.02$, the number of iterations of the denoising algorithm is reduced by 3, and the running time of the algorithm is reduced by 129 s. Similarly, when $\sigma = 0.05$, the number of iterations of the denoising algorithm is reduced by 9, and the running time is reduced by 731 s. These results show that when σ is less than 0.1, the value of SNR and MSE is almost unchanged. Therefore, in low-noise scenarios, it is advisable to first simplify the high-precision 3D cultural relic model obtained using scanning and then proceed with denoising.

Table 3. Comparison of experimental results before and after point cloud simplification on 4#yt.

Method	σ	Points	SNR(DB)	MSE	Iterations	Running Time (s)
Geometry + color	0.02	58,380	63.98	0.50	4	171
Geometry + color + simplification		30,000	64.12	0.55	1	42
Geometry + color	0.05	58,380	62.14	0.72	11	870
Geometry + color + simplification		30,000	61.98	0.76	2	139
Geometry + color	0.1	58,380	62.03	0.81	49	4011
Geometry + color + simplification		30,000	59.96	0.96	8	293
Geometry + color	0.2	58,380	54.02	1.90	106	9003
Geometry + color + simplification		30,000	51.99	1.89	29	1104

When $\sigma = 0.1$, the number of iterations of the denoising algorithm is reduced by 41, and the running time of the algorithm is reduced by 3718 s. Similarly, when $\sigma = 0.2$, the number of iterations of the denoising algorithm is reduced by 77, and the running time is reduced by 7899 s. It can be seen that the running time of the algorithm is greatly reduced, and the denoising effect is not significantly weakened when the point cloud is simplified to a reasonable size. Therefore, for some real-time application scenarios, it is necessary to simplify the point cloud before denoising.

5.3.3. Iterations

This section presents the subjective results of the proposed combined color and geometry denoising approach. Figure 7a shows the presence of numerous noise points on the surface of 4#yt, resulting in an uneven surface. Subsequently, in Figure 7b, after the third iteration, the sharp noise points on the surface of the 3D model appear smoother. Furthermore, Figure 7c shows that as the number of iterations increases to 6, the rough areas on the surface of the 3D model gradually become smoother. Finally, Figure 7d shows that when the number of iterations reaches 10, the surface noise points are effectively eliminated.

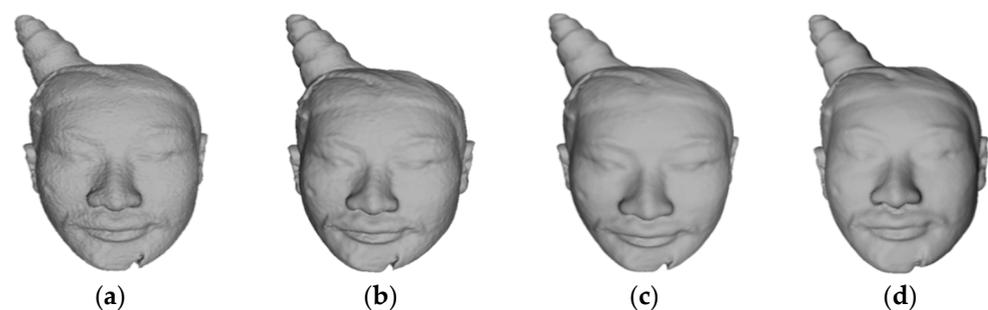


Figure 7. The denoising effect of our algorithm in different iterations on 4#yt: (a) description of the noisy input; (b) description of the third iteration; (c) description of the sixth iteration; (d) description of the tenth iteration.

The experimental results in Table 2 show that the number of iterations of the algorithm is influenced by the noise level. For $\sigma = 0.02$, the denoising algorithm needs 4 iterations; for

$\sigma = 0.05$, the denoising algorithm needs 11 iterations. It can be observed that as the noise level increases, more iterations are required. Conversely, when the noise level is low, the proposed method shows the advantages of fewer iterations and a faster convergence speed.

5.3.4. Robustness

The robustness of the proposed method was tested under different noise levels. Gaussian white noise was added to the clean 3D model in reverse, and the denoising effect of the proposed method was verified. In the experiment, the standard deviation σ of the white Gaussian noise was set to 0.02, 0.05, 0.1, and 0.2.

Figures 8 and 9 show the denoising effects of the proposed method after adding Gaussian white noise with $\sigma = 0.02$ and $\sigma = 0.05$, respectively. The experimental results indicate that at a low noise level, the surface smoothness of this method is almost indistinguishable from that of clean point clouds.

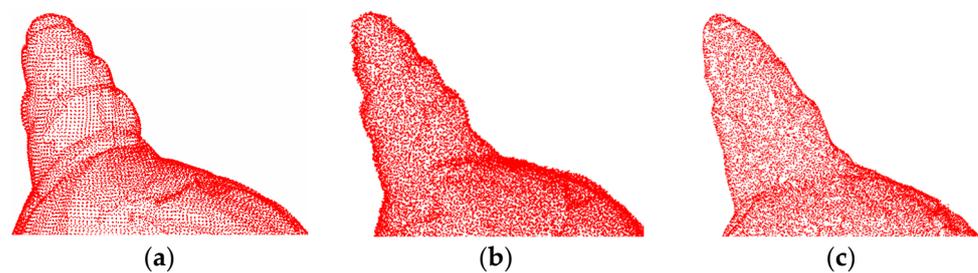


Figure 8. The denoising effect of 4#yt with Gaussian white noise with $\sigma = 0.02$ added: (a) description of the clean point cloud; (b) description of the noisy input; (c) description of the denoising effect.

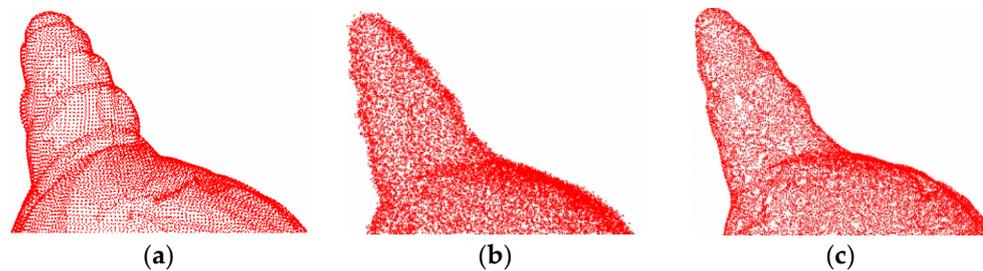


Figure 9. The denoising effect of 4#yt with Gaussian white noise with $\sigma = 0.05$ added: (a) description of the clean point cloud; (b) description of the noisy input; (c) description of the denoising effect.

Figures 10 and 11 show the denoising effects of the proposed method after adding Gaussian white noise with $\sigma = 0.1$ and $\sigma = 0.2$, respectively. Clearly, the proposed method exhibits excellent denoising performance even at high noise levels, effectively removing a significant amount of noise from the 3D point cloud surface, with only a few outliers remaining. These results demonstrate the strong robustness of the proposed method.

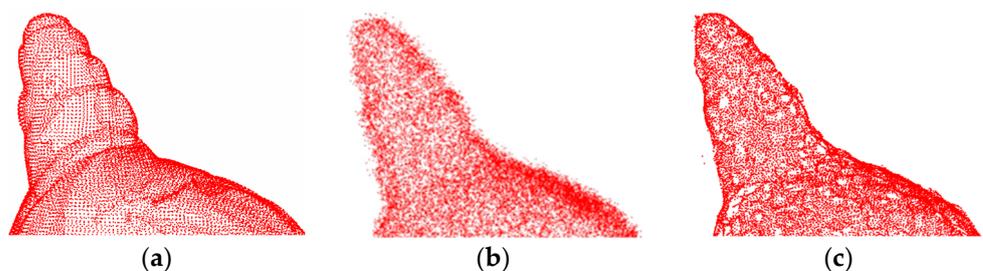


Figure 10. The denoising effect of 4#yt with Gaussian white noise with $\sigma = 0.1$ added: (a) description of the clean point cloud; (b) description of the noisy input; (c) description of the denoising effect.

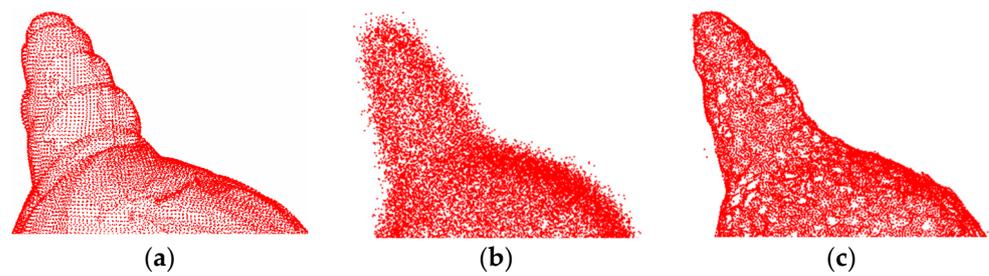


Figure 11. The denoising effect of 4#yt with Gaussian white noise with $\sigma = 0.2$ added: (a) description of clean point cloud; (b) description of the noisy input; (c) description of the denoising effect.

5.4. Comparison with Competing Methods

This section focuses on analyzing the experimental results by comparing them with other methods, both subjective and objective, using a dataset of cultural relics obtained using a 3D scanner and three public 3D point clouds. To validate the superiority of the proposed method, we compared it with MRPCA [16], LR [11], the method proposed in [31], the method proposed in [19], and the method proposed in [24] in our experiments.

5.4.1. Subjective Assessment

Figure 12 shows the denoising results using different methods on the 3D fragment numbered G10-11-43(47)4. It can be seen from Figure 12f,g that both LR [11] and the method described in [19] effectively remove surface noise from the armor of the terracotta warriors. However, these methods also result in the smoothing of sharp features such as the rivets on the armor. Figure 12d,e show that the methods mentioned in [16,31] manage to better preserve the fine features of the rivets, but the surface of the armor still remains rough and uneven, and the noise is not completely removed.

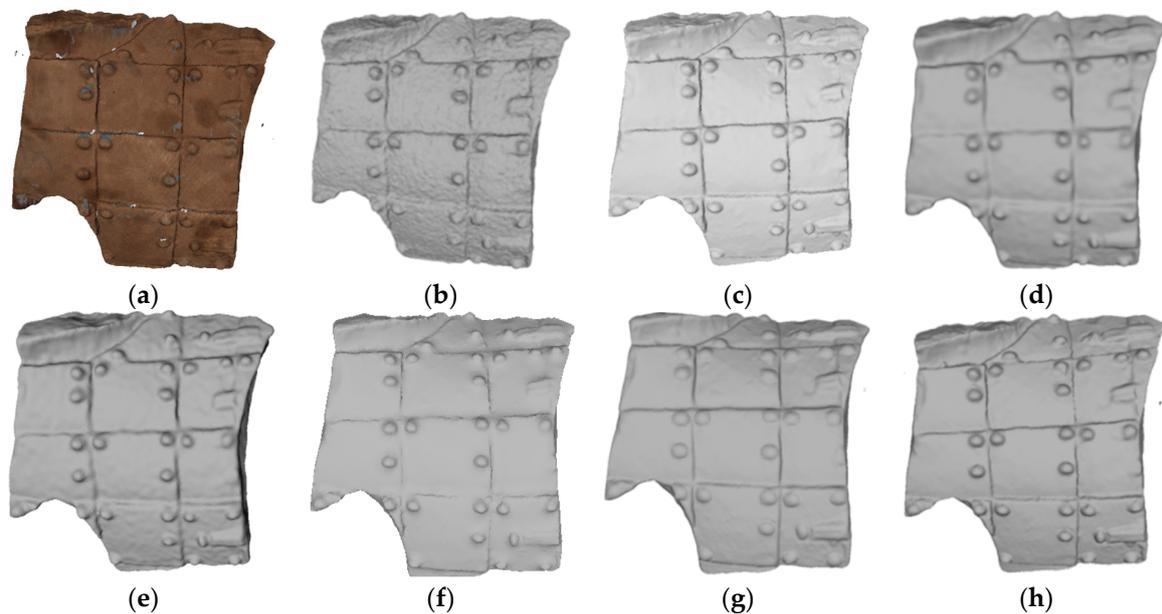


Figure 12. Denoising effect of several methods on G10-11-43(47)4: (a) description of ground truth; (b) description of noisy input; (c) description of our method; (d) description of MRPCA method; (e) description of the method in [31]; (f) description of LR method; (g) description of the method in [19]; (h) description of the method in [24].

The method proposed in [24] successfully eliminates surface noise while preserving the surface decoration of cultural relics, as shown in Figure 12h. In particular, our denoising method ensures the clear visibility of the rivets on the armor and achieves a satisfactory

smoothing effect on the model surface, as shown in Figure 12c. The resulting 3D model, after noise removal, closely resembles the real cultural relic.

The denoising effects of different methods on Q002789 are shown in Figure 13. Although the methods in Figure 13d,e can remove most of the noise on the surface of the 3D point cloud, there is still a small amount of noise attached to the surface that has not been removed. The denoising effect of the methods in Figure 13f–h is better than that of the methods in Figure 13d,e, but the pattern in the blue dotted circle is very blurred. In Figure 13c, the denoising effect of the proposed method is the most ideal and most similar to the real cultural relics, especially the area enclosed by the blue dotted circle, whose fine details are completely preserved.

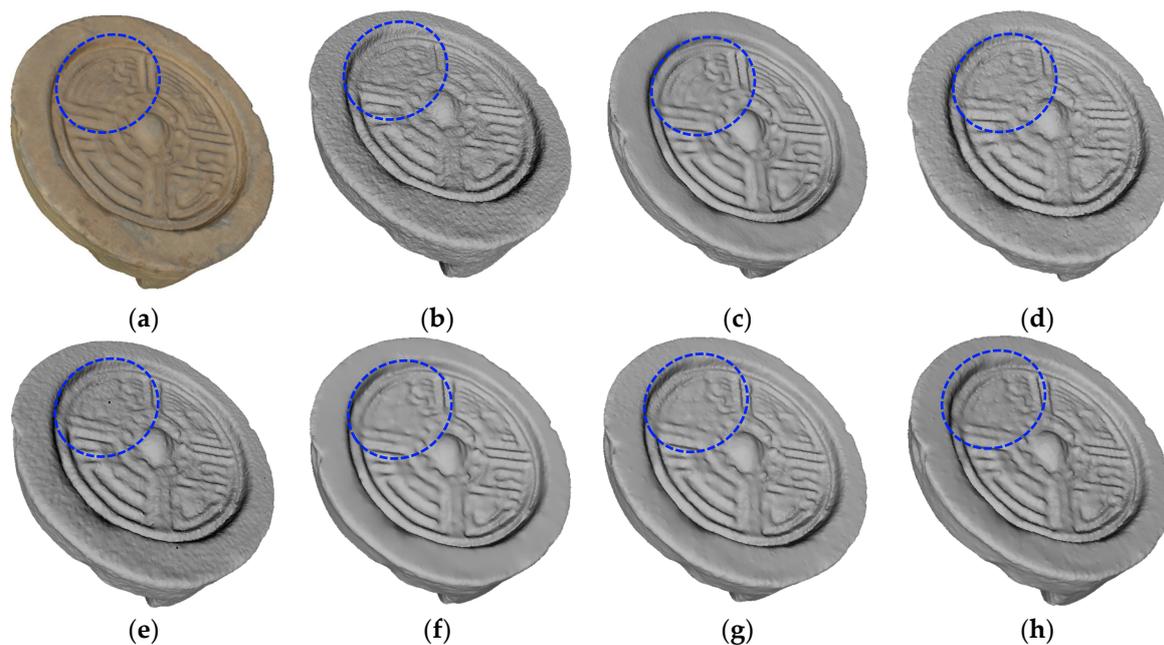


Figure 13. Denoising effect of several methods on Q002789: (a) description of the ground truth; (b) description of the noisy input; (c) description of our method; (d) description of MRPCA method; (e) description of the method in [31]; (f) description of LR method; (g) description of the method in [19]; (h) description of the method in [24].

The denoising effect of different methods on H73 is shown in Figure 14. It can be seen that, when using the denoising methods in Figure 14d,e, the surface of the sculpture of Hu terracotta army is uneven; in particular, the denoising of the face is insufficient, resulting in blurred facial contours and features. Evidently, the smoothing effect of the method in Figure 14f is better than that in Figure 14d,e. The methods described in Figure 14c,g,h overall show better performance compared to the methods in Figure 14d–f. These methods effectively enhance the clarity of facial contours and features in the Hu terracotta sculpture. For example, the details of the eyes and beards are well preserved. The method in Figure 14g is the worst of the three methods because there is still a small amount of noise on the surface that has not been completely removed. Both our method shown in Figure 14c and the method shown in Figure 14h demonstrate advanced denoising effects. It is important to highlight that the area encircled by the blue box is the hem of the dress, and several other competing methods produce unsatisfactory denoising results for this specific area, whereas our method successfully preserves the fine details of this dress.

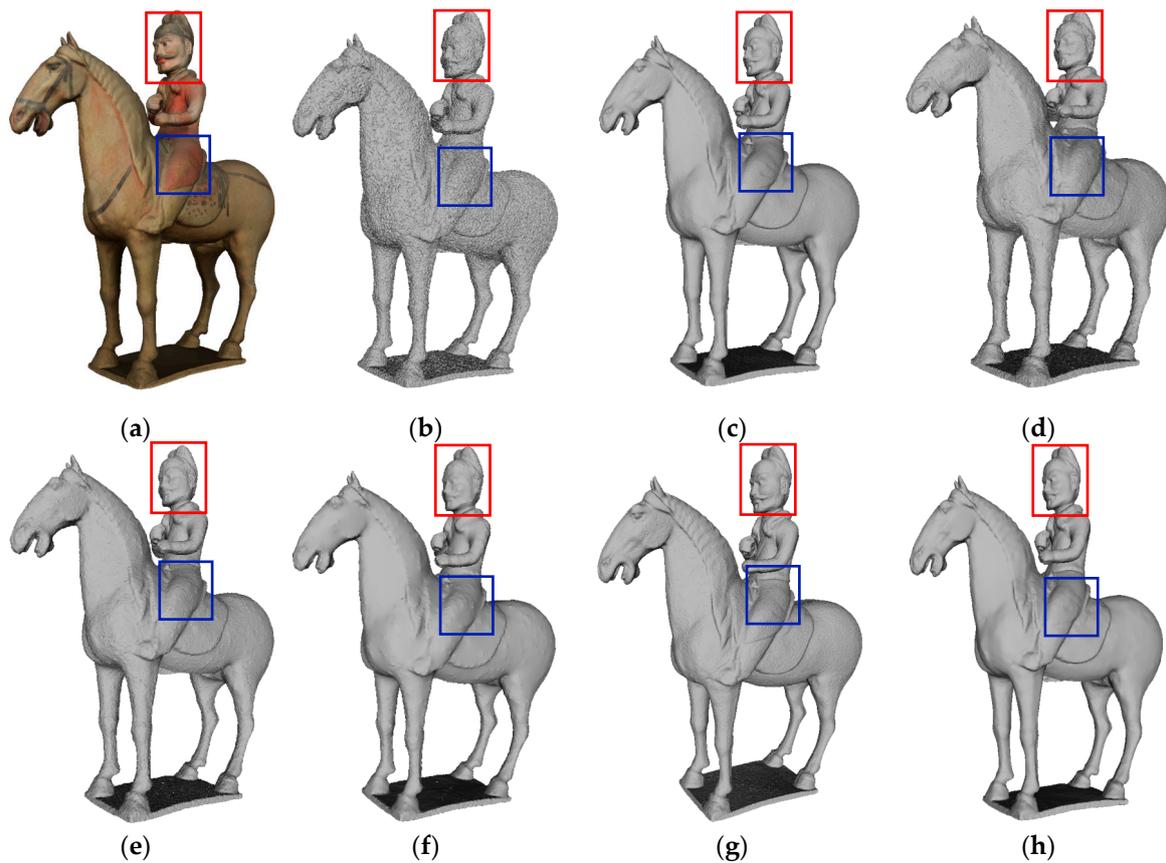


Figure 14. Denoising effect of several methods on H73: (a) description of the ground truth; (b) description of the noisy input; (c) description of our method; (d) description of MRPCA method; (e) description of the method in [31]; (f) description of LR method; (g) description of the method in [19]; (h) description of the method in [24].

The denoising results of different methods on the 3D fragment numbered G3-I-C-94 are shown in Figure 15. From Figure 15e,f, it can be seen that the finger part is too smooth after applying the denoising techniques proposed in [11,31]. Figure 15h shows that the method in [24] successfully preserves the intricate features of the palm and finger joint after noise removal. In Figure 15d,g, MRPCA [16] and the method in [19] effectively remove most of the noise, albeit with a slightly coarse denoising effect. Figure 15c shows that our proposed method is able to remove the noise substantially while still preserving the fine features of the finger cracks.

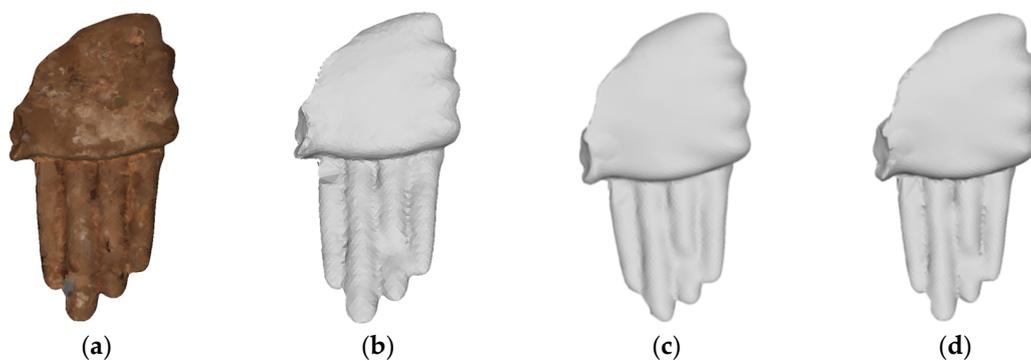


Figure 15. *Cont.*

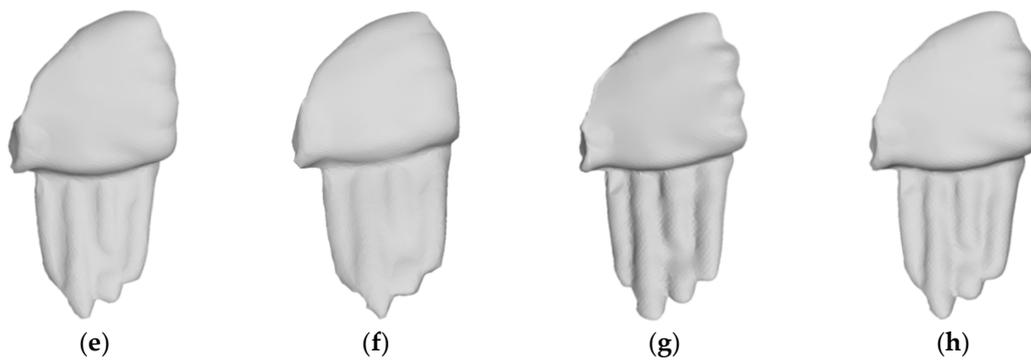


Figure 15. Denoising effect of several methods on G3-I-C-94: (a) description of ground truth; (b) description of noisy input; (c) description of our method; (d) description of MRPCA method; (e) description of the method in [31]; (f) description of LR method; (g) description of the method in [19]; (h) description of the method in [24].

5.4.2. Objective Assessment

We evaluated the proposed approach on cultural relic point clouds. To evaluate the denoising results, we introduced noise of different intensities into the clean 3D point cloud and quantitatively analyzed the results using the MSE and SNR. The experimental results in Tables 4 and 5 show that as the noise intensity increases, the mean square error between the denoised point cloud and the clean point cloud also increases. When considering the EMS or SNR, our proposed method outperforms other competing denoising methods in terms of denoising effectiveness. Furthermore, even in the presence of high-level noise, our method maintains a small deviation between the two metrics, indicating its strong robustness.

Table 4. MSE metric comparison of six methods using cultural relic data.

Data	σ	MRPCA	LR	[19]	[31]	[24]	Our Method
G10-52	0.02	0.697	0.760	0.684	0.704	0.650	0.630
G10-46-5	0.02	0.885	0.955	0.879	0.906	0.827	0.801
G3-I-C-94	0.02	0.797	0.842	0.774	0.801	0.746	0.769
G10-11-43(47)	0.02	0.658	0.678	0.633	0.659	0.613	0.573
Q002789	0.02	0.251	0.247	0.297	0.231	0.223	0.198
Q003418	0.02	0.503	0.515	0.472	0.497	0.421	0.432
H73	0.02	0.390	0.387	0.350	0.356	0.335	0.305
H80	0.02	0.567	0.581	0.549	0.545	0.534	0.521
Average		0.594	0.621	0.580	0.587	0.544	0.529
G10-52	0.03	0.801	0.826	0.799	0.819	0.759	0.733
G10-46-5	0.03	1.000	1.031	0.964	1.001	0.939	0.909
G3-I-C-94	0.03	0.940	0.948	0.899	0.921	0.873	0.826
G10-11-43(47)	0.03	0.719	0.730	0.687	0.713	0.659	0.642
Q002789	0.03	0.304	0.315	0.279	0.281	0.274	0.270
Q003418	0.03	0.579	0.576	0.561	0.570	0.554	0.542
H73	0.03	0.375	0.386	0.368	0.360	0.354	0.346
H80	0.03	0.634	0.641	0.639	0.619	0.624	0.620
Average		0.658	0.670	0.638	0.648	0.619	0.601
G10-52	0.04	0.948	0.945	0.915	0.930	0.872	0.820
G10-46-5	0.04	1.034	1.079	1.027	1.014	1.042	0.961
G3-I-C-94	0.04	1.032	1.045	0.984	1.023	0.987	0.992
G10-11-43(47)	0.04	0.741	0.734	0.717	0.722	0.699	0.641
Q002789	0.04	0.324	0.310	0.298	0.291	0.283	0.279
Q003418	0.04	0.649	0.655	0.634	0.627	0.618	0.609
H73	0.04	0.412	0.420	0.421	0.428	0.431	0.447
H80	0.04	0.749	0.740	0.759	0.733	0.711	0.723
Average		0.736	0.741	0.719	0.721	0.705	0.684

Table 4. Cont.

Data	σ	MRPCA	LR	[19]	[31]	[24]	Our Method
G10-52	0.05	0.943	1.157	0.911	0.921	0.907	0.857
G10-46-5	0.05	1.167	1.370	1.143	1.110	1.177	1.009
G3-I-C-94	0.05	1.097	1.265	1.024	1.057	1.103	0.983
G10-11-43(47)	0.05	0.867	0.948	0.833	0.802	0.756	0.709
Q002789	0.05	0.331	0.333	0.321	0.325	0.312	0.301
Q003418	0.05	0.729	0.733	0.702	0.711	0.697	0.687
H73	0.05	0.533	0.557	0.519	0.524	0.505	0.495
H80	0.05	0.802	0.823	0.785	0.799	0.791	0.779
Average		0.809	0.898	0.780	0.781	0.781	0.728
G10-52	0.1	1.158	1.188	1.199	1.149	1.045	0.941
G10-46-5	0.1	1.786	1.797	1.667	1.736	1.594	1.494
G3-I-C-94	0.1	1.433	1.453	1.338	1.421	1.281	1.105
G10-11-43(47)	0.1	1.185	1.195	1.164	1.177	1.152	0.912
Q002789	0.1	0.425	0.439	0.415	0.411	0.409	0.401
Q003418	0.1	0.749	0.750	0.736	0.741	0.730	0.724
H73	0.1	0.551	0.536	0.534	0.540	0.528	0.526
H80	0.1	1.103	1.101	1.098	1.067	1.076	0.997
Average		1.049	1.057	1.019	1.030	0.977	0.888

Table 5. SNR metric comparison of six methods using cultural relic data.

Data	σ	MRPCA	LR	[19]	[31]	[24]	Our Method
G10-52	0.02	60.10	59.53	59.86	60.99	61.29	62.33
G10-46-5	0.02	66.84	65.79	66.12	64.82	67.73	67.95
G3-I-C-94	0.02	61.03	58.55	61.85	61.42	62.28	62.24
G10-11-43(47)	0.02	66.02	65.88	66.57	65.81	67.21	67.23
Q002789	0.02	57.49	57.43	58.02	57.87	58.26	61.23
Q003418	0.02	62.68	62.70	63.74	66.34	64.54	66.21
H73	0.02	67.86	68.15	70.23	69.31	70.70	72.23
H80	0.02	71.01	69.43	71.26	71.85	73.22	73.22
Average		64.13	63.43	64.71	64.80	65.65	66.58
G10-52	0.03	58.90	58.47	59.03	58.48	59.68	59.78
G10-46-5	0.03	65.57	64.59	65.55	64.95	66.34	66.45
G3-I-C-94	0.03	59.85	59.41	59.88	59.52	61.65	61.49
G10-11-43(47)	0.03	65.54	65.46	66.02	65.83	66.51	66.99
Q002789	0.03	54.23	53.12	56.89	56.12	57.59	59.31
Q003418	0.03	60.43	60.95	61.98	62.35	62.13	62.87
H73	0.03	67.85	68.15	69.85	69.87	70.12	73.01
H80	0.03	69.27	68.99	70.56	70.04	71.23	72.01
Average		62.71	62.39	63.72	63.40	64.41	65.24
G10-52	0.04	57.79	57.37	57.64	57.93	58.25	58.41
G10-46-5	0.04	64.87	64.19	64.60	64.77	65.40	65.51
G3-I-C-94	0.04	58.80	58.77	58.95	59.01	59.37	59.32
G10-11-43(47)	0.04	65.60	64.92	65.22	65.08	66.03	66.13
Q002789	0.04	55.46	56.12	56.13	56.98	57.26	58.72
Q003418	0.04	58.57	58.66	59.02	59.54	60.03	61.23
H73	0.04	66.72	66.39	67.98	67.55	68.31	69.84
H80	0.04	66.54	66.23	66.75	67.12	67.00	67.88
Average		61.79	61.58	62.04	62.25	62.71	63.38

Table 5. Cont.

Data	σ	MRPCA	LR	[19]	[31]	[24]	Our Method
G10-52	0.05	57.14	56.87	56.75	57.08	57.82	57.96
G10-46-5	0.05	63.78	63.34	63.46	63.57	63.70	63.77
G3-I-C-94	0.05	58.45	57.67	58.25	58.13	59.05	59.11
G10-11-43(47)	0.05	64.27	63.33	64.36	64.56	65.07	65.11
Q002789	0.05	54.64	55.03	56.01	55.46	56.87	56.99
Q003418	0.05	56.44	55.85	56.77	57.03	57.98	58.34
H73	0.05	66.75	66.23	67.41	67.06	66.58	67.56
H80	0.05	63.96	63.18	64.85	64.86	64.23	65.01
Average		60.68	60.19	60.98	60.97	61.41	61.73
G10-52	0.1	54.85	54.93	54.85	54.93	55.00	55.09
G10-46-5	0.1	59.16	59.33	59.15	59.22	59.22	59.30
G3-I-C-94	0.1	55.52	55.29	55.50	55.51	55.52	55.60
G10-11-43(47)	0.1	59.55	59.43	59.54	59.48	59.57	59.66
Q002789	0.1	52.45	52.32	54.51	53.56	53.77	54.90
Q003418	0.1	53.54	53.85	53.67	53.30	53.82	53.14
H73	0.1	63.57	63.33	63.47	63.06	63.83	62.62
H80	0.1	60.86	60.23	61.77	61.66	61.29	62.34
Average		57.44	57.34	57.81	57.59	57.75	57.83

6. Conclusions

The acquisition of cultural relic point clouds can be achieved directly using 3D scanning equipment. However, this process is often imperfect, resulting in noise corruption in the point clouds. Removing noise from the surface of the cultural relic point cloud while preserving sharp details is a challenging task. To address this problem, we proposed an approach that combines color and geometric features to denoise the cultural relic point cloud. Our approach is based on graph signal processing, in which we formulated the denoising process as a minimization of graph Laplacian regularization. Utilizing color and geometric characteristics as signals, we approached the elimination of surface noise as an optimization dilemma with a graph signal smoothness prior. To evaluate the effectiveness of our denoising approach, we applied it to 3D cultural relic point clouds. It is important to highlight that the proposed approach is versatile and can be used in different applications where data are limited. The experimental results show that our approach outperforms five competing methods, effectively removing noise from the surface of cultural relic point clouds while preserving important details such as texture and ornamentation to a great extent.

Author Contributions: Conceptualization, H.G.; methodology, H.G.; software, H.W.; validation, H.G.; writing—original draft preparation, H.G.; writing—review and editing, H.G.; visualization, S.Z.; supervision, H.G.; project administration, H.G.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Ningxia Province, grant number 2022AAC03005, and the Key Research and Development Projects program of Ningxia Province, grant number 2023BDE03006.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which have improved the overall quality of this manuscript. The authors would like to thank Geng Guohua for providing the experimental data.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Liu, Z.; Song, S.; Wang, B.; Gong, W.; Ran, Y.; Hou, X.; Chen, Z.; Li, F. Multispectral LiDAR point cloud highlight removal based on color information. *Opt. Express* **2022**, *30*, 28614–28631. [CrossRef] [PubMed]
- Li, Z.Q.; Mao, X.Y.; Wang, J.; Guo, Y. Feature-preserving triangular mesh surface denoising: A survey and prospective. *J. Comput.-Aided Des. Comput. Graph.* **2020**, *32*, 1–15.
- Zhang, W.; Deng, B.; Zhang, J.; Bouaziz, S.; Liu, L. Guided mesh normal filtering. *Comput. Graph. Forum* **2015**, *34*, 23–34. [CrossRef]
- Li, N.; Yue, S.; Li, Z.; Wang, S.; Wang, H. Adaptive and feature-preserving mesh denoising schemes based on developmental guidance. *IEEE Access* **2020**, *8*, 172412–172427. [CrossRef]
- Wang, P.-S.; Fu, X.-M.; Liu, Y.; Tong, X.; Liu, S.-L.; Guo, B. Rolling guidance normal filter for geometric processing. *ACM Trans. Graph.* **2015**, *34*, 1–9. [CrossRef]
- Liu, B.; Cao, J.; Wang, W.; Ma, N.; Li, B.; Liu, L.; Liu, X. Propagated mesh normal filtering. *Comput. Graph.* **2018**, *74*, 119–125. [CrossRef]
- Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H. Edge-aware point set resampling. *ACM Trans. Graph.* **2013**, *32*, 1–12. [CrossRef]
- Zheng, Y.; Li, G.; Wu, S.; Liu, Y.; Gao, Y. Guided point cloud denoising via sharp feature skeletons. *Vis. Comput.* **2017**, *33*, 857–867. [CrossRef]
- Sun, Y.; Schaefer, S.; Wang, W. Denoising point sets via L_0 minimization. *Comput. Aided Geom. Des.* **2015**, *35*, 2–15. [CrossRef]
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.* **2009**, *28*, 1–7. [CrossRef]
- Sarkar, K.; Bernard, F.; Varanasi, K.; Theobalt, C.; Stricker, D. Structured low-rank matrix factorization for point-cloud denoising. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 444–453. [CrossRef]
- Li, X.; Zhu, L.; Fu, C.; Heng, P. Non-local low-rank normal filtering for mesh denoising. *Comput. Graph. Forum* **2018**, *37*, 155–166. [CrossRef]
- Osher, S.; Shi, Z.; Zhu, W. Low dimensional manifold model for image processing. *SIAM J. Imaging Sci.* **2017**, *10*, 1669–1690. [CrossRef]
- Avron, H.; Sharf, A.; Greif, C.; Cohen-Or, D. L_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph.* **2010**, *29*, 1–12. [CrossRef]
- Zhao, Y.; Li, L.; Shan, X.; Wang, S.; Qin, S.; Wang, H. A L_0 denoising algorithm for 3D shapes. *J. Comput.-Aided Des. Comput. Graph.* **2018**, *30*, 772–777.
- Mattei, E.; Castrodad, A. Point cloud denoising via moving RPCA. *Comput. Graph. Forum* **2017**, *36*, 123–137. [CrossRef]
- Han, X.-F.; Jin, J.S.; Wang, M.-J.; Jiang, W.; Gao, L.; Xiao, L. A review of algorithms for filtering the 3D point cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [CrossRef]
- Gao, X.; Hu, W.; Tang, J.; Liu, J.; Guo, Z. Optimized skeleton-based action recognition via sparsified graph regression. In Proceedings of the 27th ACM International Conference on Multimedia, New York, NY, USA, 21–25 October 2019; pp. 601–610.
- Dinesh, C.; Cheung, G.; Bajic, I.V. Point cloud denoising via feature graph laplacian regularization. *IEEE Trans. Image Process.* **2020**, *29*, 4143–4158. [CrossRef] [PubMed]
- Shang, X.; Ye, R.; Feng, H.; Jiang, X. Robust Feature Graph for Point Cloud Denoising. In Proceedings of the 7th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, 18–20 November 2022.
- Egilmez, H.E.; Pavez, E.; Ortega, A. Graph learning from data under laplacian and structural constraints. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 825–841. [CrossRef]
- Jiang, B.; Zhang, Z.Y.; Lin, D.D.; Tang, J.; Luo, B. Semi-supervised learning with graph learning-convolutional networks. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
- Hu, W.; Gao, X.; Cheung, G.; Guo, Z. Feature graph learning for 3D point cloud denoising. *IEEE Trans. Signal Process.* **2020**, *68*, 2841–2856. [CrossRef]
- Hu, W.; Hu, Q.; Wang, Z.; Gao, X. Dynamic point cloud denoising via manifold-to-manifold distance. *IEEE Trans. Image Process.* **2021**, *30*, 6168–6183. [CrossRef]
- Hu, W.; Pang, J.; Liu, X.; Tian, D.; Lin, C.W.; Vetro, A. Graph signal processing for geometric data and beyond: Theory and applications. *IEEE Trans. Multimed.* **2022**, *24*, 3961–3977. [CrossRef]
- Zeng, J.; Cheung, G.; Ng, M.; Pang, J.; Yang, C. 3D point cloud denoising using graph laplacian regularization of a low dimensional manifold model. *IEEE Trans. Image Process.* **2020**, *29*, 3474–3489. [CrossRef]
- Wang, J.; Huang, J.; Wang, F.L.; Wei, M.; Xie, H.; Qin, J. Data-driven geometry-recovering mesh denoising. *Comput. Des.* **2019**, *114*, 133–142. [CrossRef]
- Zhao, W.B.; Liu, X.M.; Zhao, Y.S.; Fan, X.P.; Zhao, D.B. NormalNet: Learning based guided normal filtering for mesh denoising. *arXiv* **2019**, arXiv:1903.04015v2. Available online: <https://arxiv.org/abs/1903.04015v2> (accessed on 4 November 2019).
- Li, Z.; Pan, W.; Wang, S.; Tang, X.; Hu, H. A point cloud denoising network based on manifold in an unknown noisy environment. *Infrared Phys. Technol.* **2023**, *132*, 104735. [CrossRef]

30. Huang, A.; Xie, Q.; Wang, Z.; Lu, D.; Wei, M.; Wang, J. MODNet: Multi-offset point cloud denoising network customized for multi-scale patches. *Comput. Graph. Forum* **2022**, *41*, 109–119. [[CrossRef](#)]
31. Cattai, T.; Delfino, A.; Scarano, G.; Colonnese, S. VIPDA: A visually driven point cloud denoising algorithm based on anisotropic point cloud filtering. *Front. Signal Process.* **2022**, *2*, 842570. [[CrossRef](#)]
32. Hu, X.; Wei, X.; Sun, J. A noising-denoising framework for point cloud upsampling via normalizing flows. *Pattern Recognit. J. Pattern Recognit. Soc.* **2023**, *140*, 109569. [[CrossRef](#)]
33. Liu, Y.; Sheng, H.K. A single-stage point cloud cleaning network for outlier removal and denoising. *Pattern Recognit.* **2023**, *138*, 109366. [[CrossRef](#)]
34. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **2013**, *30*, 83–98. [[CrossRef](#)]
35. Li, X.; Han, J.; Yuan, Q.; Zhang, Y.; Fu, Z.; Zou, M.; Huang, Z. FEUSNet: Fourier Embedded U-Shaped Network for Image Denoising. *Entropy* **2023**, *25*, 1418. [[CrossRef](#)]
36. Li, R.Z.; Yang, M.; Ran, Y.; Zhang, H.H.; Jing, J.F.; Li, P.F. Point cloud denoising and simplification algorithm based on method library. *Laser Optoelectron. Prog.* **2018**, *55*, 251–257.
37. Liu, Y.; Sun, Y. Laser point cloud denoising based on principal component analysis and surface fitting. *Laser Technol.* **2020**, *44*, 103–108.
38. Yang, D.; Sun, J. BM3D-Net: A convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Process. Lett.* **2017**, *25*, 55–59. [[CrossRef](#)]
39. Wei, X.; van Gorp, H.; Carabarin, L.G.; Freedman, D.; Eldar, Y.C.; van Sloun, R.J.G. Image denoising with deep unfolding and normalizing flows. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022.
40. Wei, X.; Van Gorp, H.; Gonzalez-Carabarin, L.; Freedman, D.; Eldar, Y.C.; van Sloun, R.J. Deep unfolding with normalizing flow priors for inverse problems. *IEEE Trans. Signal Process.* **2022**, *70*, 2962–2971. [[CrossRef](#)]
41. Dutta, S.; Basarab, A.; Georgeot, B.; Kouame, D. Quantum mechanics-based signal and image representation: Application to denoising. *IEEE Open J. Signal Process.* **2021**, *2*, 190–206. [[CrossRef](#)]
42. Dutta, S.; Basara, A.; Georgeot, B.; Kouamé, D. A novel image denoising algorithm using concepts of quantum many-body theory. *Signal Process.* **2022**, *201*, 108690. [[CrossRef](#)]
43. Shi, Z.; Xu, W.; Meng, H. A point cloud simplification algorithm based on weighted feature indexes for 3D scanning sensors. *Sensors* **2022**, *22*, 7491. [[CrossRef](#)]
44. Pasqualotto, G.; Zanuttigh, P.; Cortelazzo, G.M. Combining color and shape descriptors for 3D model retrieval. *Signal Process. Image Commun.* **2013**, *28*, 608–623. [[CrossRef](#)]
45. Musicco, A.; Galantucci, R.A.; Bruno, S.; Verdoscia, C.; Fatiguso, F. Automatic point cloud segmentation for the detection of alterations on historical buildings through an unsupervised and clustering-based machine learning approach. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *2*, 129–136. [[CrossRef](#)]
46. Vinodkumar, P.K.; Karabulut, D.; Avots, E.; Ozcinar, C.; Anbarjafari, G. A Survey on Deep Learning Based Segmentation, Detection and Classification for 3D Point Clouds. *Entropy* **2023**, *25*, 635. [[CrossRef](#)]
47. Irfan, A.M.; Magli, E. Exploiting color for graph-based 3D point cloud denoising. *J. Vis. Commun. Image Represent.* **2021**, *75*, 103027. [[CrossRef](#)]
48. Rosman, G.; Dubrovina, A.; Kimmel, R. Patch-collaborative spectral point-cloud denoising. *Comput. Graph. Forum* **2013**, *32*, 1–12. [[CrossRef](#)]
49. Barkalov, K.; Shtanyuk, A.; Sysoyev, A. A Fast kNN Algorithm Using Multiple Space-Filling Curves. *Entropy* **2022**, *24*, 767. [[CrossRef](#)] [[PubMed](#)]
50. Miranda-González, A.A.; Rosales-Silva, A.J.; Mújica-Vargas, D.; Escamilla-Ambrosio, P.J.; Gallegos-Fune, F.J.; Vianney-Kinani, J.M.; Velázquez-Lozada, E.; Pérez-Hernández, L.M.; Lozano-Vázquez, L.V. Denoising Vanilla Autoencoder for RGB and GS Images with Gaussian Noise. *Entropy* **2023**, *25*, 1467. [[CrossRef](#)] [[PubMed](#)]
51. De Gregorio, J.; Sánchez, D.; Toral, R. Entropy Estimators for Markovian Sequences: A Comparative Analysis. *Entropy* **2024**, *26*, 79. [[CrossRef](#)] [[PubMed](#)]
52. Paige, C.; Saunders, M. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Trans. Math. Softw.* **1982**, *8*, 43–71. [[CrossRef](#)]
53. Parikh, N.; Boyd, S. Proximal Algorithms. *Found. Trends Optim.* **2013**, *1*, 123–231.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.