

Article

Denoising for 3D Point Cloud Based on Regularization of a Statistical Low-Dimensional Manifold

Youyu Liu ^{1,2,†} , Baozhu Zou ^{1,2,*}, Jiao Xu ^{3,†}, Siyang Yang ^{1,2} and Yi Li ^{1,2}

¹ Key Laboratory of Advanced Perception and Intelligent Control of High-End Equipment, Ministry of Education, Wuhu 241000, China; liuyyu@ahpu.edu.cn (Y.L.); 2012124@ahpu.edu.cn (S.Y.); 2210110112@stu.ahpu.edu.cn (Y.L.)

² School of Mechanical Engineering, Anhui Polytechnic University, Wuhu 241000, China

³ System Office, Wuhu Changxin Technology Co., Ltd., Wuhu 241009, China; jxu@tokengroup.com

* Correspondence: 2200110117@stu.ahpu.edu.cn

† These authors contributed equally to this work.

Abstract: A point cloud obtained by stereo matching algorithm or three-dimensional (3D) scanner generally contains much complex noise, which will affect the accuracy of subsequent surface reconstruction or visualization processing. To eliminate the complex noise, a new regularization algorithm for denoising was proposed. In view of the fact that 3D point clouds have low-dimensional structures, a statistical low-dimensional manifold (SLDM) model was established. By regularizing its dimensions, the denoising problem of the point cloud was expressed as an optimization problem based on the geometric constraints of the regularization term of the manifold. A low-dimensional smooth manifold model was constructed by discrete sampling, and solved by means of a statistical method and an alternating iterative method. The performance of the denoising algorithm was quantitatively evaluated from three aspects, i.e., the signal-to-noise ratio (SNR), mean square error (MSE) and structural similarity (SSIM). Analysis and comparison of performance showed that compared with the algebraic point-set surface (APSS), non-local denoising (NLD) and feature graph learning (FGL) algorithms, the mean SNR of the point cloud denoised using the proposed method increased by 1.22 DB, 1.81 DB and 1.20 DB, respectively, its mean MSE decreased by 0.096, 0.086 and 0.076, respectively, and its mean SSIM decreased by 0.023, 0.022 and 0.020, respectively, which shows that the proposed method is more effective in eliminating Gaussian noise and Laplace noise in common point clouds. The application cases showed that the proposed algorithm can retain the geometric feature information of point clouds while eliminating complex noise.

Keywords: point cloud denoising; statistical low-dimensional manifold; regularization; Gaussian noise; Laplace noise



Citation: Liu, Y.; Zou, B.; Xu, J.; Yang, S.; Li, Y. Denoising for 3D Point Cloud Based on Regularization of a Statistical Low-Dimensional Manifold. *Sensors* **2022**, *22*, 2666. <https://doi.org/10.3390/s22072666>

Academic Editor: Anastasios Doulamis

Received: 22 January 2022

Accepted: 26 March 2022

Published: 30 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Due to their simplicity, flexibility and strong representative ability, three-dimensional (3D) point clouds are more and more widely used in many fields [1–3], such as object recognition and surface reconstruction. Point cloud data are mainly obtained in two ways [4,5], namely using a stereo matching algorithm or a 3D scanner. The former obtains the point cloud by projecting all pixels into the 3D space through matching [6]. Due to the fuzzy matching, a lot of noise will be present in the obtained point cloud. The latter, including laser scanners [7], structured light scanners [8] and lidar [9], can quickly obtain complete point clouds. However, due to the uncertainties of measurement errors, reflectance from objects, occlusion, illumination and the environment, the obtained point cloud data of objects often contain a large number of complex noise points [10]. Noise not only deforms the bottom manifold structure of point clouds [11], which is not conducive to their surface reconstruction and visualization, but also adds useless information [12], and then reduces the accuracy of the extraction of their features.

Point cloud denoising has become one of the hot fields of 3D geometric data processing. Zeng et al. [13] extended a previously proposed graph Laplace regularization (GLR), which uses the patch manifold prior to seeking self-similar patches in order to fulfill the purpose of point cloud denoising. Dinesh et al. [14] proposed a fast graph-based local algorithm, but it was found that the graph signal denoising algorithm would lead to the problems of clustering and deformation. Duan et al. [15] proposed a weighted multi-projection (WMP) algorithm to estimate a local tangent plane at each 3D point, and then reconstructed each 3D point through the weighted averaging of its projections on multiple tangent planes, so as to alleviate the problems of clustering and deformation. Wei et al. [16] applied the feature graph learning (FGL) algorithm to point cloud denoising. Different from setting the edge weight parameters manually, when the available signal is smooth relative to the graph, the edge weight parameters are strictly optimized through feature measurement learning, so as to realize the accurate and fast denoising of the point cloud. The above algorithms are derived from formulae that explicitly assume Gaussian noise, and have achieved good results regarding the additional noise generated on the object surface, but theoretically, they are not suitable for the actual complex noise far away from the object noise points. The removal of actual complex noise points is still challenging.

To solve the issue of the actual complex noise of point clouds in objects, a denoising approach for 3D point clouds, based on the regularization of the statistical low-dimensional manifold (R-SLDM), is proposed in this article. Firstly, the slight and non-sparse noise (such as Gaussian noise) and large and sparse noise (such as Laplace noise) (see Figure 1) were combined into actual complex noise. Secondly, in view of the fact that 3D point clouds have low-dimensional structures, the point cloud model with noise was distributed on the low-dimensional manifold in a high-dimensional space. Moreover, the regularization term of the manifold was used for the denoising of prior information to maintain the basic structure embedded in the low-dimensional manifold. Finally, the denoising problem of point clouds was expressed as an optimization problem based on the geometric constraints of the regularization term of the manifold, which was approximately solved by means of a statistical method and an alternating iterative method.

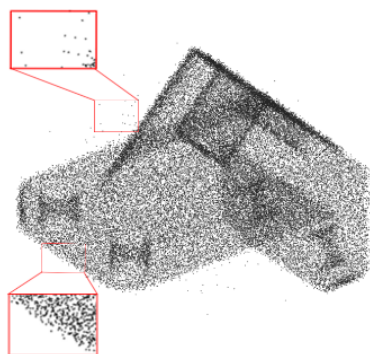


Figure 1. Point cloud model with Laplace noise and Gaussian noise.

The main contributions of our work are as follows:

- (1) A denoising algorithm was designed for Gaussian noise and Laplace noise to denoise the two kinds of noise together;
- (2) Using the regularization term of the manifold and the fidelity term of the noise, the basic structure of a denoised point cloud was maintained;
- (3) Discrete sampling was used to construct low-dimensional manifolds to avoid a large number of calculations.

2. Related Works

At present, there are two kinds of denoising methods for point cloud data: local methods [14,17–25] and non-local methods [13,15,16,26–28]. The former involves the denoising

of the point cloud based on the neighborhood of points, while the latter involves identifying similar point patches from the local neighborhood and combining this group of patches for denoising.

(1) Local Method. Alexa et al. [17] defined a point cloud surface by means of moving least-squares projection, projected the noise points onto the least-squares surface, and then smoothed the data. Guennebaud et al. [18] and Öztireli et al. [19] proposed the algebraic point set surface (APSS) and robust implicit moving least-squares (RIMLS), respectively, for denoising based on moving least-squares (MLS), but it was found that over-smoothing could easily be produced [20,21]. Lipman et al. [22] introduced local optimal projection (LOP) in point cloud denoising, of which the basic principle is to project the point set to the natural surface of the point cloud to reduce noise. However, the algorithm has a limited effect in terms of preserving the features of the point cloud. Huang et al. [23] proposed a weighted local optimal projection (WLOP) algorithm based on the local optimal projection algorithm, which can retain the features of a point cloud, but it is time-consuming. Moreover, they also proposed an anisotropic WLOP (AWLOP) algorithm by modifying WLOP, and used anisotropic weighting functions to retain the sharp features of the point cloud model [24]; however, it can easily produce additional features [20,21]. Dinesh et al. [25] solved different types of additive noise by using two fidelity terms through feature graph Laplace regularization (FGLR).

(2) Non-Local Method. Dabov et al. [26] and Rosman et al. [27] spread image denoising algorithms such as non-local mean (NLM) and block-matching 3D Denoising (BM3D) to point cloud denoising. However, this was found to depend on the self-similarity among the surface blocks in the point cloud, and required a large amount of calculation. Deschaud et al. [28] used the polynomial coefficients of local MLS surfaces as neighborhood descriptors to calculate the similarity of points, and then proposed a non-local denoising (NLD) algorithm. Zeng et al. [13] proposed the GLR denoising algorithm to seek self-similar patches to denoise the point cloud. Different from the direct smoothing of 3D coordinate points, Duan et al. [15] estimated the local tangent plane of each point based on the graphical method, and realized point cloud denoising through weighted projection. Wei et al. [16] used the FGL algorithm for point cloud denoising.

The method proposed in this article is categorized as a non-local method. Similar to [13,16,25], we also used regularization for denoising. However, the methods given in references [13,16] involve the establishing of prior information based on Gaussian noise, which is not suitable for complex information including both Laplace noise and Gaussian noise, so it is difficult to apply these methods in practical denoising. On the other hand, although two kinds of additive noise are studied in reference [25], two different algorithms are designed for them; thus, the two kinds of noise are not solved uniformly. Our proposed method is intended to solve the above two problems.

3. Methods

3.1. Low-Dimensional Manifold Model

Image $\mathbf{u} \in \mathbb{R}^{m \times n}$ (indicating that \mathbf{u} is in the number field with the size of $m \times n$); $\mathcal{P}(\mathbf{u})(x)$ is defined as a two-dimensional (2D) rectangular pixel block of the \mathbf{u} , with a size of $s_1 \times s_2$. Pixel x is in the upper left corner of the rectangle, and $\forall x \in \bar{\Omega} = \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$. The set of all pixel blocks is called the pixel block set of the \mathbf{u} , which is represented by $P(\mathbf{u})$ [29]:

$$P(\mathbf{u}) = \{\mathcal{P}(\mathbf{u})(x) : x \in \bar{\Omega}\} \subset \mathbb{R}^d, d = s_1 \times s_2 \quad (1)$$

For the image \mathbf{u} , a point cloud in \mathbb{R}^d is given from the $P(\mathbf{u})$, which is close to a smooth manifold embedded in \mathbb{R}^d . This potential smooth manifold is called a block manifold; it is associated with \mathbf{u} , and is denoted as $\mathcal{M}(\mathbf{u})$. Based on the fact that there are often

low-dimensional structures in many natural image block manifolds [29,30], Osher et al. designed a low-dimensional manifold model [31], as shown in Equation (2):

$$\arg \min_{\substack{\mathbf{u} \in \mathbb{R}^{m \times n} \\ \mathcal{M} \subset \mathbb{R}^d}} \dim(\mathcal{M}), \text{ subject to : } \mathbf{b} = \Phi \mathbf{u} + \varepsilon, P(\mathbf{u}) \subset \mathcal{M} \quad (2)$$

Moreover, $\dim(\mathcal{M})$ in Equation (2) is expressed as follows:

$$\dim(\mathcal{M}) = \sum_{i=1}^d \|\nabla_{\mathcal{M}} \alpha_i(\mathbf{x})\|_{L^2(\mathcal{M})}^2 \quad (3)$$

where $\alpha_i(\mathbf{x}) = x_i; \forall \mathbf{x} = (x_1, \dots, x_d) \in \mathcal{M} \subset \mathbb{R}^d$

An image restoration model for low-dimensional manifolds is obtained from Equations (2) and (3), as shown in Equation (4):

$$\arg \min_{\substack{\mathbf{u} \in \mathbb{R}^{m \times n} \\ \mathcal{M} \subset \mathbb{R}^d}} \sum_{i=1}^d \|\nabla_{\mathcal{M}} \alpha_i(\mathbf{x})\|_{L^2(\mathcal{M})}^2 + \mu \|\mathbf{b} - \Phi \mathbf{u}\|_2^2, \text{ subject to : } P(\mathbf{u}) \subset \mathcal{M} \quad (4)$$

3.2. Point Cloud Denoising Model

3.2.1. 3D Point Cloud and Noise Model

Point cloud $\mathcal{N} = \{\mathbf{n}_i\}_{i=1}^N$, and $\mathbf{n}_i \in \mathbb{R}^3$. Let $\mathbf{N} = [\mathbf{n}_1, \dots, \mathbf{n}_N]^T \in \mathbb{R}^{N \times 3}$, and then its observation model can be expressed as Equation (5):

$$\mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L} \quad (5)$$

where $\mathbf{U}, \mathbf{G}, \mathbf{L} \in \mathbb{R}^{N \times 3}$.

Our goal in developing this model was to restore the observation model \mathbf{N} to the ideal model \mathbf{U} under the interference of Gaussian noise \mathbf{G} and Laplace noise \mathbf{L} .

3.2.2. Statistical Low-Dimensional Manifold Model

A surface block $\{\mathbf{n}_m\}_{m=1}^M \subset \mathcal{N}$ is defined in the point cloud, from which M subsets are selected as the block centers. The block set p_m centered on \mathbf{n}_m is defined as the set of k points closest to \mathbf{n}_m , and $U_M^{m=1} p_m = \mathcal{N}$. Let $\mathbf{p}_m \in \mathbb{R}^{3k}$ be the block coordinates, composed of k points in p_m .

A low-dimensional smooth manifold, in which p_m samples are embedded in \mathbb{R}^{3k} , is represented by $\mathcal{M}(\mathbf{U})$. The designed statistical low-dimensional manifold (SLDM) model is shown in Equation (6):

$$\arg \min_{\mathbf{U}} \dim(\mathcal{M}(\mathbf{U})), \text{ subject to : } \mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L}, p_m \subset \mathcal{M}(\mathbf{U}) \quad (6)$$

The maximum a posteriori (MAP) problem can be described by a regularization term and a fidelity term, as follows:

$$\arg \min_{\mathbf{U}} \dim(\mathcal{M}(\mathbf{U})) + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2, \text{ subject to : } p_m \subset \mathcal{M}(\mathbf{U}) \quad (7)$$

where $\|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2$ is the fidelity item, which is used to ensure the similarity before and after point cloud processing; $\min_{\mathbf{U}} \dim(\mathcal{M}(\mathbf{U}))$ is the regularization term to enhance the output; λ is a parameter to weigh the relationship between the fidelity term and the regularity term.

Blocks are usually not a single smooth manifold, which may have different dimensions and correspond to different patterns of images [29]; thus, $\dim(\mathcal{M}(\mathbf{U}))$ of the block manifold

is a function of $\mathcal{M}(\mathbf{U})$. The integral of $\dim(\mathcal{M}(\mathbf{U}))$ on \mathcal{M} is used for regularization, as shown in Equation (8):

$$\arg \min_{\mathbf{U}} \int_{\mathcal{M}} \dim(\mathcal{M}(\mathbf{U}))(\mathbf{p}) d\mathbf{p} + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2, \text{ subject to: } p_m \subset \mathcal{M}(\mathbf{U}) \quad (8)$$

where $\mathbf{p} \in \mathbb{R}^{3k}$ is a point on \mathcal{M} ; $\dim(\mathcal{M}(\mathbf{U}))(\mathbf{p})$ is the manifold size of $\mathcal{M}(\mathbf{U})$ at \mathbf{p} . According to Equation (3),

$$\dim(\mathcal{M})(\mathbf{p}) = \sum_{i=1}^{3k} \|\nabla_{\mathcal{M}} f_i(\mathbf{p})\|_F^2 \quad (9)$$

Substituting Equation (9) into Equation (8),

$$\arg \min_{\mathbf{U}} \sum_{i=1}^{3k} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f_i(\mathbf{p})\|_F^2 d\mathbf{p} + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2, \text{ subject to: } \mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L} \quad (10)$$

where $\forall \mathbf{p} = [p_1, \dots, p_{3k}]^T \in \mathcal{M} \subset \mathbb{R}^{3k}$, $f_i(\mathbf{p}) = p_i$; $\nabla_{\mathcal{M}} f_i(\mathbf{p})$ is the gradient of function f_i at \mathbf{p} on \mathcal{M} .

3.3. Solution of Point Cloud Denoising Model

3.3.1. Solution Principle

Manifold \mathcal{M} is discretely sampled to construct a discrete graph \mathcal{G} , and to describe the low-dimensional smooth manifold model. Its vertex set is a visible surface block $\mathcal{P} = \{\mathbf{p}_m\}_{m=1}^M$, and $\mathbf{p}_m \in \mathcal{M}(\mathbf{U}) \subset \mathbb{R}^{3k}$, in which the edge weight of the m and n blocks are defined as follows:

$$w_{m,n} = \begin{cases} (\rho_m \rho_n)^{-1/\gamma} \exp\left(-\frac{\|\mathbf{p}_m - \mathbf{p}_n\|_2^2}{2\epsilon^2}\right) & \|\mathbf{p}_m - \mathbf{p}_n\|_2 < r \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where $(\rho_m \rho_n)^{-1/\gamma}$ is a normalized term; $\rho_n = \sum_{m=1}^M \psi(\|\mathbf{p}_m - \mathbf{p}_n\|_2)$ is the expression before normalization; $\psi(\cdot)$ is the weight kernel function. Under these conditions, an r -neighborhood graph is constructed with no edge being greater than r ; $r = \epsilon C_r$.

A symmetric adjacency matrix $\mathbf{W} \in \mathbb{R}^{M \times M}$ is defined by using the edge weight in Equation (11) to represent graph \mathcal{G} , in which $\mathbf{W}(m, n) = \mathbf{W}(n, m) = w_{m,n}$. Moreover, $\mathbf{D}(m, m) = \sum_n w_{m,n}$. According to the literature [32], the Laplacian matrix of the combined graph is defined as $\mathbf{M} = \mathbf{D} - \mathbf{W}$. For the regularization term in Equation (10), f_i is sampled at the position of \mathcal{P} to obtain the discrete form $f_i = [f_i(\mathbf{p}_1) \dots f_i(\mathbf{p}_M)]^T$. The regularizer $f_i^T \mathbf{M} f_i$ [33] is derived from \mathbf{M} , as shown in Equation (12).

$$f_i^T \mathbf{M} f_i = \sum_{(m,n) \in \tau} w_{m,n} (f_i(\mathbf{p}_m) - f_i(\mathbf{p}_n))^2 \quad (12)$$

According to studies [34,35],

$$\lim_{\substack{M \rightarrow \infty \\ \epsilon \rightarrow 0, \delta \rightarrow 0}} f_i^T \mathbf{M} f_i \sim \frac{1}{|\mathcal{M}|} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f_i(\mathbf{p})\|_F^2 d\mathbf{p} \quad (13)$$

With the increasing of sample number M and the decreasing of neighborhood radius r , $f_i^T \mathbf{M} f_i$ is close to its limit of smoothness. If the manifold dimension δ is low, even if the block manifold is embedded in a high-dimensional space, it can reasonably approximate the continuous regularization function [36].

According to Equations (9) and (13),

$$\lim_{\substack{M \rightarrow \infty \\ \epsilon \rightarrow 0, \delta \rightarrow 0}} |\mathcal{M}| \sum_{i=1}^{3k} f_i^\top \mathbf{M} f_i \sim \sum_{i=1}^{3k} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f_i(\mathbf{p})\|_2^2 d\mathbf{p} \quad (14)$$

In order to define f_i , the $\{\mathbf{p}_m\}_{m=1}^M$ must be sorted. The following Equation (15) can be inferred from Equation (12):

$$\sum_{i=1}^{3k} f_i^\top \mathbf{M} f_i = \sum_{(m,n) \in \tau} w_{m,n} \|\mathbf{p}_m - f_i \mathbf{p}_n\|_2^2 \quad (15)$$

where $\|\mathbf{p}_m - \mathbf{p}_n\|_2$ can determine the block similarity. It can be seen from Equation (15) that the regularization does not need the coordinate function f_i ; thus, the surface blocks do not need to be sorted.

According to the constructed discrete graph \mathcal{G} , the global graph Laplace matrix $\mathbf{M}_p = \sum_{i=1}^{3k} \mathbf{S}^T \mathbf{M} \mathbf{S} \in \mathbb{R}^{kM \times N}$ can be obtained, in which kM is the number of points of all surface blocks, and $\mathbf{S} \in \{0, 1\}^{kM \times N}$. Therefore, Equation (10) can be transformed into Equation (16):

$$\arg \min_{\mathbf{U}} \left(\mathbf{P}_x^\top \mathbf{M}_p \mathbf{P}_x + \mathbf{P}_y^\top \mathbf{M}_p \mathbf{P}_y + \mathbf{P}_z^\top \mathbf{M}_p \mathbf{P}_z \right) + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2, \text{ subject to : } \mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L} \quad (16)$$

where $\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z \in \mathbb{R}^{kM}$.

Let $\mathbf{P} = [\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z] \in \mathbb{R}^{kM \times 3}$, then,

$$\text{tr}(\mathbf{P}^\top \mathbf{M}_p \mathbf{P}) = \mathbf{P}_x^\top \mathbf{M}_p \mathbf{P}_x + \mathbf{P}_y^\top \mathbf{M}_p \mathbf{P}_y + \mathbf{P}_z^\top \mathbf{M}_p \mathbf{P}_z \quad (17)$$

where \mathbf{P} is related to the denoised point cloud sample, and $\mathbf{P} = \mathbf{S}\mathbf{U} - \mathbf{C}$, $\mathbf{C} \in \mathbb{R}^{kM \times 3}$.

The objective function (16) can be rewritten as shown in Equation (18):

$$\arg \min_{\mathbf{U}} \text{tr} \left((\mathbf{S}\mathbf{U} - \mathbf{C})^\top \mathbf{M}_p (\mathbf{S}\mathbf{U} - \mathbf{C}) \right) + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2, \text{ subject to : } \mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L} \quad (18)$$

For any point \mathbf{n}_i in the point cloud \mathcal{N} , let S_i represent the average distance from this point to k points in the neighborhood, and then the distance threshold is obtained, as shown in Equation (19):

$$d_{\text{threshold}} = \frac{\sum_{i=1}^n S_i}{N} \pm \text{std} \cdot \frac{\sum_{i=1}^n \left(S_i - \frac{\sum_{i=1}^n S_i}{N} \right)^2}{N} \quad (19)$$

The point with a distance between \mathbf{n}_i and a neighborhood outside $\left(\frac{\sum_{i=1}^n S_i}{N} - \text{std} \cdot \frac{\sum_{i=1}^n \left(S_i - \frac{\sum_{i=1}^n S_i}{N} \right)^2}{N}, \frac{\sum_{i=1}^n S_i}{N} + \text{std} \cdot \frac{\sum_{i=1}^n \left(S_i - \frac{\sum_{i=1}^n S_i}{N} \right)^2}{N} \right)$ is Laplace noise. Therefore, the position matrix \mathbf{L} of Laplace noise can be obtained using a statistical method.

To solve Equation (18) approximately, an alternating optimization method is adopted: in the loop, fix \mathbf{M}_p and update \mathbf{U} , then update \mathbf{M}_p given \mathbf{U} , and repeat until convergence. In each iteration, \mathbf{M}_p is updated according to the above method. In order to fix \mathbf{M}_p and optimize \mathbf{U} , the coordinates (x, y, z) of each point are given by Equation (20):

$$\left(\mathbf{S}^\top \mathbf{M}_p \mathbf{S} + \lambda \mathbf{E}_q \right) \mathbf{U}_q = \lambda (\mathbf{N}_q + \mathbf{L}_q) + \mathbf{S}^\top \mathbf{M}_p \mathbf{C}_q \quad (20)$$

where \mathbf{E}_q is an identity matrix with the same size as \mathbf{M}_p ; \mathbf{U} is solved by means of an iterative method until the result converges.

3.3.2. Algorithm Design

Based on the solution principle, the denoising process for the 3D point cloud, based on the regularization of SLDM (R-SLDM), is shown in Figure 2, and its algorithm (Algorithm 1) is shown as follows:

Algorithm 1: Denoising for 3D point cloud based on R-

Input: $\mathbf{N}, k, s, \lambda, \zeta$.

Output: Denoised cloud \mathbf{U} .

1: Initializing \mathbf{U} with \mathbf{N} ;

2: **for** $iter = 1, 2, \dots, r$ **do**;

3: Sampling s points from \mathbf{U} as a block center;

4: Find the k nearest neighbors of the center of each block to form a surface block;

5: Optimizing objective function: $\arg \min_{\mathbf{U}} \text{tr} \left((\mathbf{S}\mathbf{U} - \mathbf{C})^T \mathbf{M}_p (\mathbf{S}\mathbf{U} - \mathbf{C}) \right) + \lambda \|\mathbf{N} - \mathbf{U} - \mathbf{L}\|_F^2$,

subject to: $\mathbf{N} = \mathbf{U} + \mathbf{G} + \mathbf{L}$;

6: $\mathbf{L} \leftarrow d_{\text{threshold}} = \frac{\sum_{i=1}^n S_i}{N} \pm \text{std} \cdot \frac{\sum_{i=1}^n \left(S_i - \frac{\sum_{i=1}^n S_i}{N} \right)^2}{N}$;

7: $\mathbf{M}_p \leftarrow \sum_{i=1}^{3k} \mathbf{S}^T \mathbf{M} \mathbf{S}$;

8: $\mathbf{N}_q \leftarrow \left(\mathbf{S}^T \mathbf{M}_p \mathbf{S} + \lambda \mathbf{E}_q \right) \mathbf{U}_q = \lambda (\mathbf{N}_q + \mathbf{L}_q) + \mathbf{S}^T \mathbf{M}_p \mathbf{C}_q$; $q \in \{x, y, z\}$;

9: \mathbf{U} converges;

10: **end for**

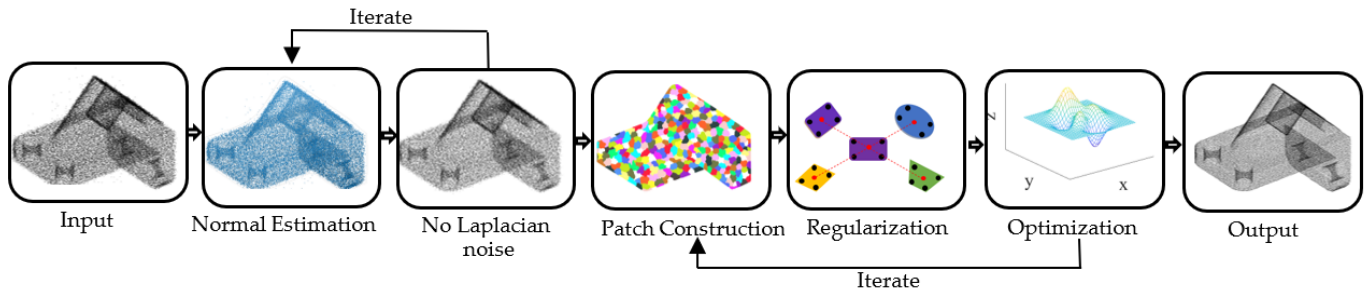


Figure 2. The flowchart of the proposed point cloud denoising algorithm.

3.3.3. Performance Evaluation

The performance of the denoising algorithm can be quantitatively evaluated based on the signal-to-noise ratio (SNR), mean square error (MSE) and structural similarity (SSIM).

The SNR [13] is shown in Equation (21):

$$\text{SNR}(\mathcal{U}_1, \mathcal{U}_2) = 10 \lg \frac{\frac{2}{N_2} \sum_{\mathbf{u}_i \in \mathcal{U}_2} \|\mathbf{u}_j\|_2^2}{\frac{1}{N_1} \sum_{\mathbf{u}_i \in \mathcal{U}_1} \min_{\mathbf{u}_j \in \mathcal{U}_2} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \frac{1}{N_2} \sum_{\mathbf{u}_i \in \mathcal{U}_2} \min_{\mathbf{u}_j \in \mathcal{U}_1} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2} \quad (21)$$

The MSE [15] is shown in Equation (22):

$$\text{MSE}(\mathcal{U}_1, \mathcal{U}_2) = \frac{1}{2} \left(\frac{1}{N_1} \sum_{\mathbf{u}_i \in \mathcal{U}_1} \min_{\mathbf{u}_j \in \mathcal{U}_2} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \frac{1}{N_2} \sum_{\mathbf{u}_i \in \mathcal{U}_2} \min_{\mathbf{u}_j \in \mathcal{U}_1} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right) \quad (22)$$

where $\mathcal{U}_1 = \{\mathbf{u}_i\}_{i=1}^{N_1}$, $\mathbf{u}_i \in \mathbb{R}^3$; $\mathcal{U}_2 = \{\mathbf{u}_j\}_{j=1}^{N_2}$, $\mathbf{u}_j \in \mathbb{R}^3$.

The SSIM [37] is shown in Equation (23):

$$\text{SSIM}(\mathcal{U}_1, \mathcal{U}_2) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)} \quad (23)$$

$$\text{where } \mu_1 = \sum_{\mathbf{u}_i \in \mathcal{U}_1} w_i |\mathbf{u}_i|, \mu_2 = \sum_{\mathbf{u}_j \in \mathcal{U}_2} w_j |\mathbf{u}_j|, \sigma_1 = \left[\sum_{\mathbf{u}_i \in \mathcal{U}_1} w_i (|\mathbf{u}_i - \mu_1|)^2 \right]^{\frac{1}{2}},$$

$$\sigma_2 = \left[\sum_{\mathbf{u}_j \in \mathcal{U}_2} w_j (|\mathbf{u}_j - \mu_2|)^2 \right]^{\frac{1}{2}}, \sigma_{12} = \frac{1}{2} \left[\sum_{\mathbf{u}_i \in \mathcal{U}_1} w_i (|\mathbf{u}_i - \mu_1|) (|\mathbf{u}_j - \mu_2|) + \sum_{\mathbf{u}_j \in \mathcal{U}_2} w_j (|\mathbf{u}_j - \mu_2|) (|\mathbf{u}_i - \mu_1|) \right], \sum_{i=1}^N w_i = 1.$$

With the SNR, MSE and SSIM, the difference between the ground-truth point cloud and the denoised one can be reasonably compared. The larger the SNR and SSIM, the lower the distortion of point cloud denoising, and the better the denoising effect. The smaller the MSE value, the lower the deviation between the ground-truth value and the denoised cloud, and the better the denoising effect.

4. Results and Discussion

4.1. Comparative Analysis of Denoising Performance

To verify the effectiveness of the proposed algorithm, the public point cloud model known as Daratech was used to carry out the denoising research, and the performance was compared with the existing algorithms under different noise intensity conditions.

As shown in Figure 3a, Daratech's ground-truth point cloud model contains 32,003 data points. The Gaussian noise and Laplacian noise, with a mean value of 0 and standard deviations (σ) of 0.1, 0.2, 0.3, 0.4, 0.5, were added to the model of the ground-truth point cloud, for which the noise models with $\sigma = 0.2$ and $\sigma = 0.4$ are shown in Figure 4. To conveniently display the effect and denoising results after adding noise, color information was added to the resulting model, and the color bar represents depth information. The R-SLDM proposed in this article was used to denoise the 3D point cloud model under different noise intensities, as shown in Figure 4, in which k is 16, s is 50%, λ is 21 and ζ is 15. From the denoising results in Figure 3b,c, it can be seen that the denoising algorithm proposed in this article can effectively eliminate Gaussian noise and Laplace noise in the common point cloud, and can properly retain their original geometric features. Thus, it is an effective point cloud denoising algorithm.

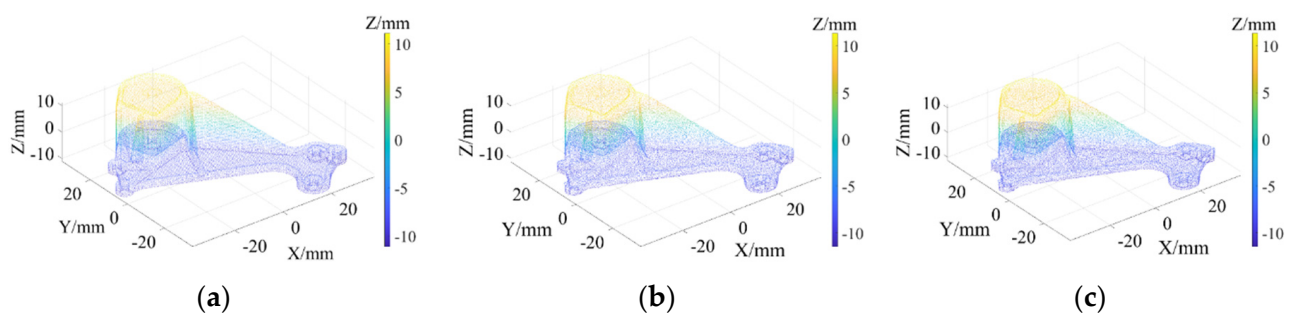


Figure 3. Ground-truth point cloud and the denoising results by R-SLDM under different noise intensities: (a) ground-truth; (b) R-SLDM ($\sigma = 0.2$); (c) R-SLDM ($\sigma = 0.4$).

The point cloud models under different noise intensities, as shown in Figure 4, were denoised by APSS, NLD and FLD, respectively. It can be seen from Figure 5 that the color distribution on the surface of the point cloud model was uneven, and thus, did not show the plane characteristics well. Moreover, there was noise far away from the main body. Compared with Figure 5, the color distribution on the surfaces shown in Figures 6 and 7 was relatively uniform, but there was also noise far away from the main body. It can be seen from the results of Figures 5–7 that the Laplace noise could not be well eliminated. As can be seen from Table 1, for the five kinds of noise levels with different σ values (0.01 to 0.05), the mean SNRs of APSS and NLD after denoising were 46.79 DB and 46.42 DB, respectively, with little difference. The mean SNR of the latest FGL method after denoising

was 47.03 DB, while that of the proposed algorithm in this article was 48.23 DB, which was 1.22 DB more than that of APSS, 1.81 DB more than that of NLD and 1.20 DB more than that of FGL. It was thus better than the latter three. The distortion of the denoising process of the proposed algorithm was the lowest. It can be seen from Table 2 that the mean MSE obtained by the R-SLDM in this article was as low as 0.159, 0.096 lower than that of APSS, 0.086 lower than that of NLD, and 0.076 lower than that of FLD. The deviation between the ground-truth value and the denoised cloud was the lowest. It can be seen from Table 3 that the mean SSIM with noise was 0.906, and those of APSS, WLOP and FGL were 0.944, 0.945 and 0.947, respectively, with all of these playing a certain role in the composite noise. However, the mean SSIM of the algorithm in this article was 0.967, which was 0.023, 0.022 and 0.020 more than that of the other three algorithms. Comparative experiments showed that the proposed algorithm is more suitable for denoising noise in practice.

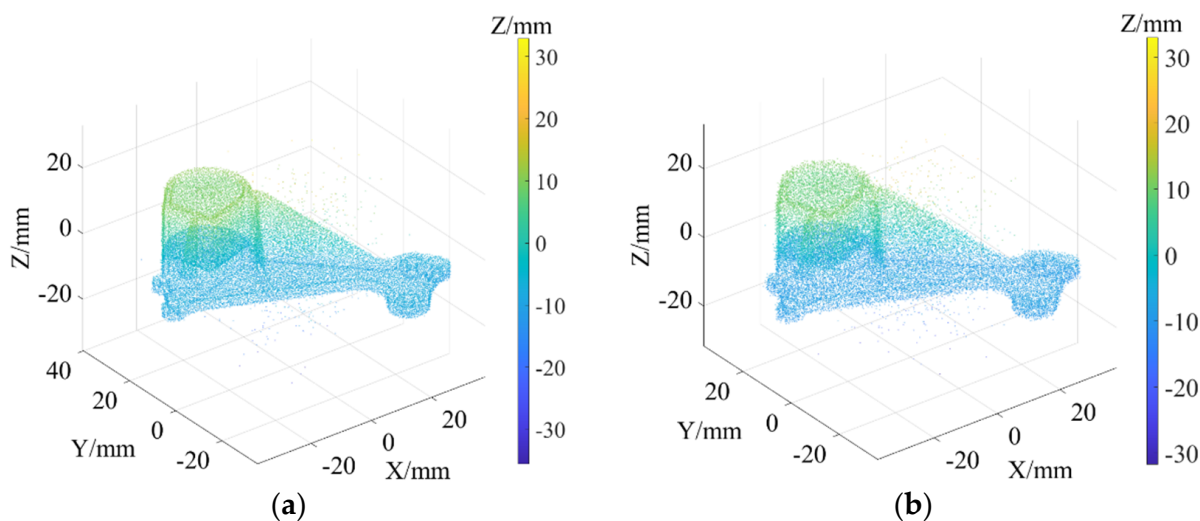


Figure 4. Point cloud models under different noise intensities: (a) noise ($\sigma = 0.2$); (b) noise ($\sigma = 0.4$).

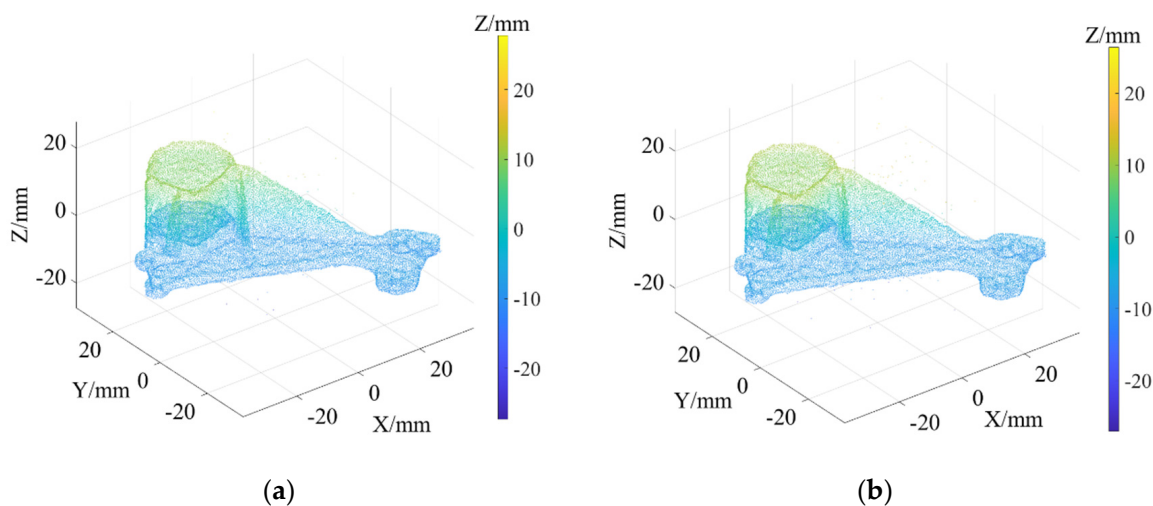


Figure 5. Denoising results of APSS under different noise intensities: (a) APSS ($\sigma = 0.2$); (b) APSS ($\sigma = 0.4$).

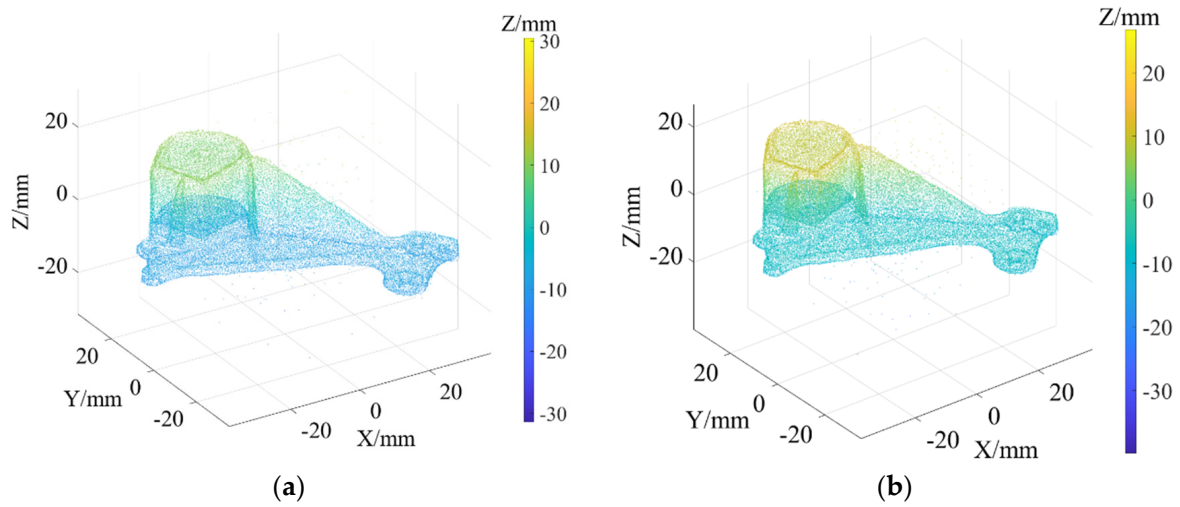


Figure 6. Denoising results of NLD under different noise intensities: (a) NLD ($\sigma = 0.2$); (b) NLD ($\sigma = 0.4$).

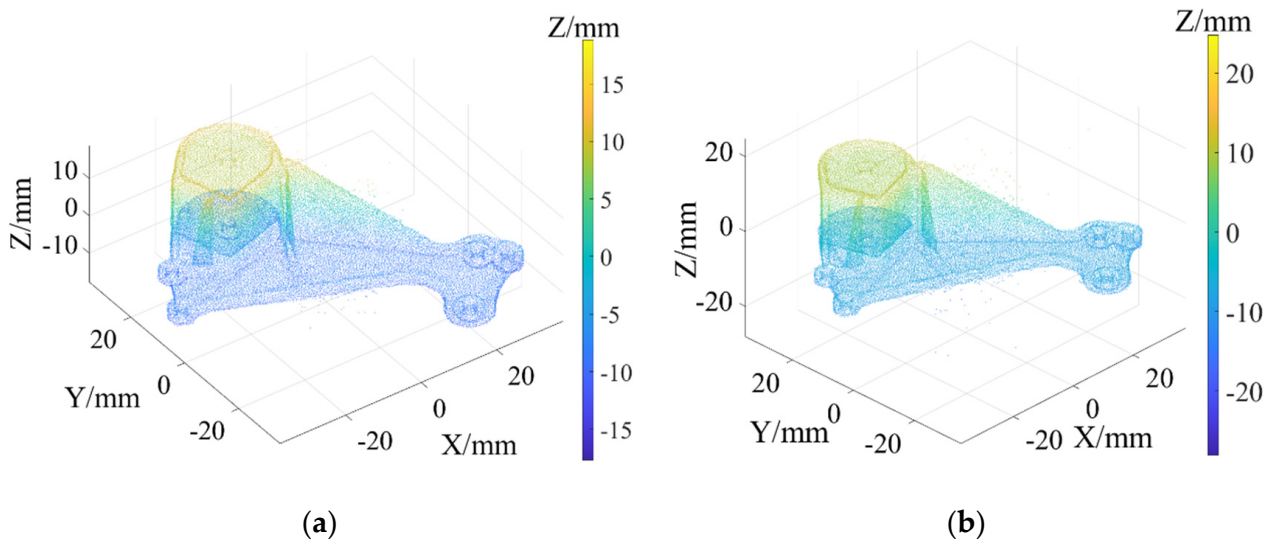


Figure 7. Denoising results of FGL under different noise intensities: (a) FGL ($\sigma = 0.2$); (b) FGL ($\sigma = 0.4$).

Table 1. SNR (DB) of five different denoising algorithms for Daratech models.

Noise Levels	$\sigma = 0.01$	$\sigma = 0.02$	$\sigma = 0.03$	$\sigma = 0.04$	$\sigma = 0.05$	Means
Noise	47.31	46.73	46.14	45.26	44.96	46.08
APSS	48.12	47.52	46.87	45.96	45.48	46.79
NLD	47.93	46.28	46.71	45.78	45.38	46.42
FGL	48.05	48.29	46.22	46.75	45.83	47.03
R-SLDM	49.78	48.77	48.26	47.36	46.98	48.23

Table 2. MSE of five different denoising algorithms for Daratech models.

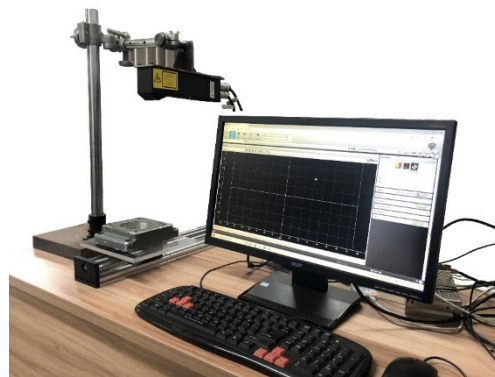
Noise Levels	$\sigma = 0.01$	$\sigma = 0.02$	$\sigma = 0.03$	$\sigma = 0.04$	$\sigma = 0.05$	Means
Noise	0.196	0.231	0.283	0.315	0.356	0.276
APSS	0.187	0.216	0.267	0.289	0.313	0.255
WLOP	0.179	0.208	0.257	0.273	0.308	0.245
FGL	0.174	0.192	0.236	0.274	0.301	0.235
R-SLDM	0.139	0.153	0.162	0.169	0.172	0.159

Table 3. SSIM of five different denoising algorithms for Daratech models.

Noise Levels	$\sigma = 0.01$	$\sigma = 0.02$	$\sigma = 0.03$	$\sigma = 0.04$	$\sigma = 0.05$	Means
Noise	0.944	0.921	0.913	0.895	0.856	0.906
APSS	0.954	0.936	0.967	0.949	0.913	0.944
WLOP	0.959	0.948	0.957	0.931	0.928	0.945
FGL	0.981	0.953	0.931	0.937	0.931	0.947
R-SLDM	0.979	0.983	0.962	0.961	0.952	0.967

4.2. Denoising Application for 3D Point Cloud

To further verify the proposed R-SLDM model, a self-built laser-scanning platform, as shown in Figure 8, was used to collect the point cloud of the four objects with noise, and the algorithm in this article was used for denoising on a workstation. The workstation parameters were as follows: the basic frequency was 3.90 GHz; the CPU model was AMD Ryzen73800X; and the graphics card model was 3060.

**Figure 8.** Laser-scanning platform.

The objects shown in Figure 9 were scanned by laser, and the obtained 3D point cloud is shown in Figure 10. To clearly illustrate the point cloud noise of objects and the denoising performance of various algorithms, we locally amplified the scanned point cloud and denoising results. It can be seen from the local enlarged view of the point clouds of the four objects in Figure 10 that there were regular noises close to the main bodies and irregular noises far away from them in the point clouds obtained by the laser-scanning platform. The results obtained for the denoising of the point cloud with the APSS algorithm are shown in Figure 11. Taking object 3 as an example, it is obvious from Figures 10c and 11c that there was little difference before and after point cloud denoising. Consequently, the APSS algorithm could not eliminate the noise far away from the main body, and the removal effect of noise on the object surface was also not effective. The results obtained for the denoising of the point cloud shown in Figure 10 using the NLD algorithm are shown in Figure 12. As can be seen from Figure 12a,b, the noise on the object surface was reduced considerably, but the noise far away from the main body still existed. As can be seen from Figure 12c,d, noise still existed on the surface of the workpiece and away from the main body. By analyzing the structural information of the four objects, it can be seen that the structures of objects 1 and 2 were simple with regular planes or surfaces, and those of objects 3 and 4 were complex, containing a large number of irregular holes and surfaces. It was found that the NLD algorithm was not suitable for the point clouds with complex structures and that were far away from the main body. The denoising results obtained with the FGL algorithm are shown in Figure 13. The noise on the surface of the four objects was well eliminated, and the noise far away from the main body was not removed. The results obtained for the denoising of the point cloud shown in Figure 10 using the proposed R-SLDM algorithm are shown in Figure 14, in which k is 16, s is 50%, λ is 21 and ξ is 15. The results show that the surfaces of the objects were smooth and the noise away from the main body became less.

Compared with APSS, NLD and FGL, the R-SLDM algorithm proposed in this article was able to effectively remove large-scale irregular noise far from the main body and small-scale regular noise close to the main body. Taking object 1 in Table 4 as an example, the SNR value of the R-SLDM algorithm was 47.83 db, which was 2.62 DB higher than that of APSS, 1.08 DB higher than that of NLD and 0.91 DB higher than that of FLD. The SNR value of the proposed R-SLDM algorithm was higher than that of APSS, NLD and FGL. For different objects, the mean SNR of R-SLDM was 48.49 DB, which was 1.02 DB higher than that of APSS, 1.14 DB higher than that of NLD and 0.94 DB higher than that of FGL. From Table 5 it can be inferred that the mean MSE of the proposed R-SLDM algorithm was reduced to 3.165×10^{-3} , a lower value than that of APSS (1.213×10^{-3}), NLD (1.107×10^{-3}) and FGL (0.451×10^{-3}). According to Table 6, the mean SSIM values of APSS, WLOP and FGL were 0.935, 0.937 and 0.950, respectively. All of the three algorithms maintained the structure of the point cloud. Nevertheless, the mean SSIM of the proposed algorithm was 0.973, which was 0.038, 0.036 and 0.023 more than that of the other three algorithms, and it retained the real structure of the point cloud more completely. Comparative experiments showed that the R-SLDM algorithm not only eliminated complex noise, but also effectively retained the geometric information of the point cloud, and the distortion rate was low.

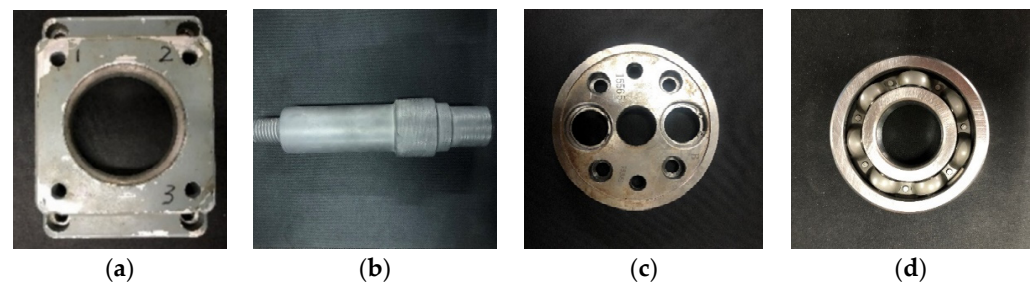


Figure 9. Scanned objects: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

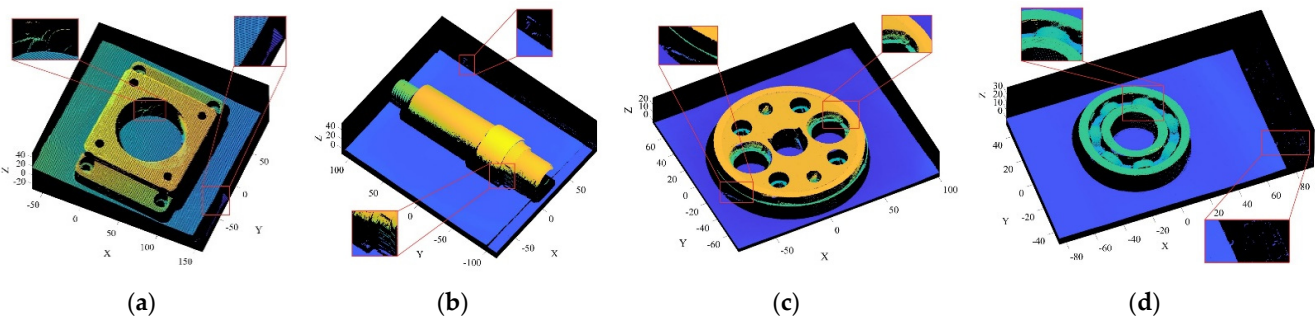


Figure 10. Point cloud models scanned by laser: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

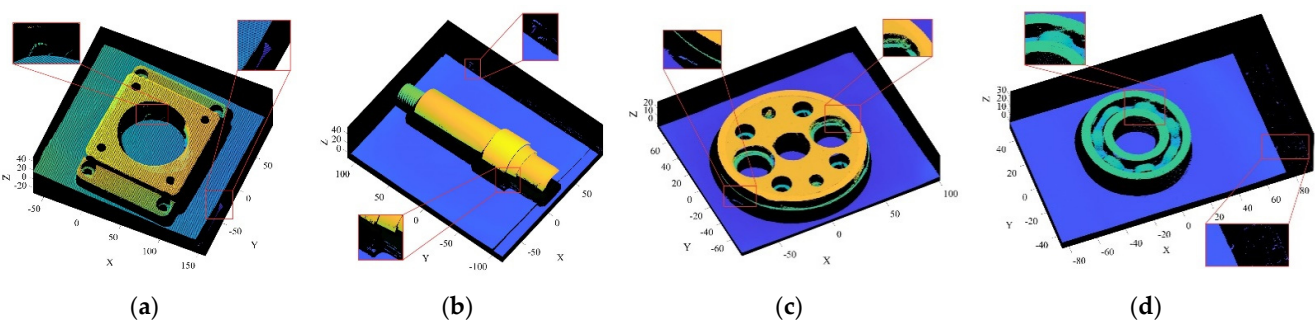


Figure 11. Denoising results for APSS algorithm: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

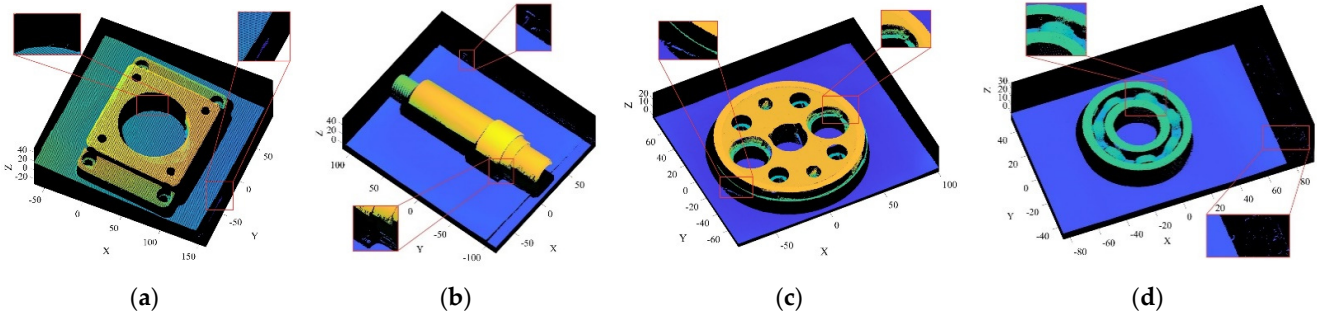


Figure 12. Denoising results for NLD algorithm: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

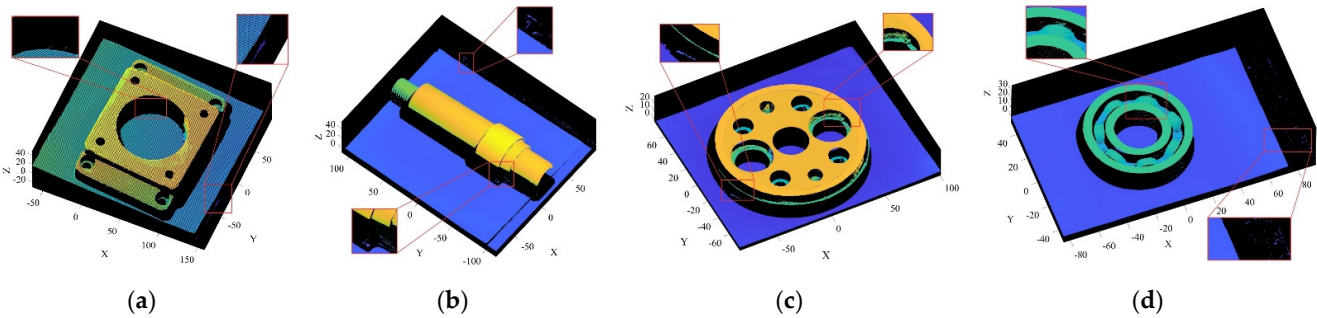


Figure 13. Denoising results for FGL algorithm: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

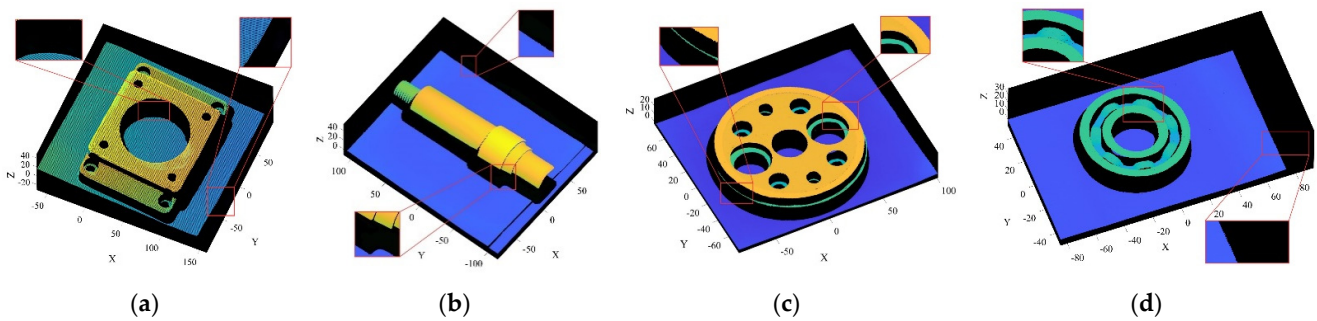


Figure 14. Denoising results for R-SLDM algorithm: (a) object 1; (b) object 2; (c) object 3; (d) object 4.

Table 4. SNR (DB) of different denoising algorithms for four objects.

Objects	a	b	c	d	Means
Noise	45.21	47.51	48.42	44.23	46.34
APSS	46.63	48.63	48.98	45.63	47.47
NLD	46.75	47.74	48.78	46.14	47.35
FGL	46.92	47.82	48.83	47.81	47.85
R-SLDM	47.83	48.64	49.74	47.73	48.49

Table 5. MSE ($\times 10^{-3}$) of different denoising algorithms for four objects.

Objects	a	b	c	d	Means
Noise	4.734	4.348	5.378	5.134	4.989
APSS	4.257	3.789	4.898	4.568	4.378
WLOP	4.191	3.695	4.788	4.414	4.272
FGL	3.933	2.978	3.764	3.789	3.616
R-SLDM	3.275	2.784	3.356	3.246	3.165

Table 6. SSIM of different denoising algorithms for four objects.

Objects	a	b	c	d	Means
Noise	0.884	0.912	0.878	0.934	0.902
APSS	0.913	0.941	0.918	0.968	0.935
WLOP	0.931	0.965	0.908	0.944	0.937
FGL	0.956	0.953	0.928	0.954	0.950
R-SLDM	0.975	0.984	0.956	0.976	0.973

5. Conclusions

A SLDM model was established in this paper, which was sampled discretely. The position matrix of the ideal point cloud was approximately calculated using the statistical method and an alternating iterative method; then, the denoising for the 3D point cloud, based on R-SLDM, which can be widely applied to the denoising of point clouds with significant geometric characteristics, was determined. Different algorithms were used to denoise the Daratech model with five kinds of noise levels, and SNR, MES and SSIM were used for comparative analysis. Compared with APSS, NLD and FGL, the mean SNR of the point cloud denoised by the proposed algorithm increased by 1.22 DB, 1.81 DB and 1.20 DB, respectively; the mean MSE decreased by 0.096, 0.086 and 0.076, respectively; the mean SSIM increased by 0.023, 0.022 and 0.020, respectively. Different algorithms were used to denoise the 3D noisy point clouds of the four kinds of objects, which shows that compared with APSS, NLD and FGL, the mean SNR of the point cloud denoised by the proposed algorithm increased by 2.62 DB, 1.08 DB and 0.91 DB, respectively; the mean MSE decreased by 1.213×10^{-3} , 1.107×10^{-3} and 0.451×10^{-3} , respectively; and the mean SSIM increased by 0.038, 0.036 and 0.023, respectively. The results show that the proposed method is more suitable for denoising practical complex noise than the existing algorithms, and can clearly retain the salient features of the objects without excessive smoothing. The method proposed in this article has achieved great success in static point cloud denoising, but has not been verified in the field of dynamic point clouds. In future work, we will consider the noise caused by irregular sampling and the change of the number of points in each frame of the dynamic point cloud. The dynamic point cloud sequence can be divided into each frame independently, and then the proposed method can be extended to the field of dynamic point clouds with strengthening of the inter-frame correlation.

Author Contributions: Conceptualization, Y.L. (Youyu Liu) and B.Z.; methodology, Y.L. (Youyu Liu); software, B.Z.; validation, J.X., B.Z. and Y.L. (Yi Li); formal analysis, S.Y.; investigation, J.X.; resources, J.X.; data curation, B.Z.; writing—original draft preparation, B.Z.; writing—review and editing, Y.L. (Youyu Liu); visualization, J.X.; supervision, B.Z.; project administration, Y.L. (Youyu Liu); funding acquisition, Y.L. (Youyu Liu). All authors have read and agreed to the published version of the manuscript.

Funding: Supported in part by the Special Fund for Collaborative Innovation of Anhui Polytechnic University & Jiujiang District under Grant No. 2021CYXTB3, by the Natural Science Research Project of Higher Education of Anhui Province of China under Grant No. KJ2020A0357, and by the Natural Science Research Project of Higher Education of Anhui Province of China under Grant No. YJS20210446.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Nomenclature

b	observation image
C	position matrix of block center
C_r, C_1, C_2	constant
d	number
$\dim(\mathcal{M})$	dimension of block manifold \mathcal{M}
$d_{\text{threshold}}$	distance threshold
D	diagonal matrix
f_i	coordinate function
G	Gaussian noise
i	count
k	number of nearest fields
L	Laplace noise
M	Laplacian matrix of combined graph
\mathbf{M}_p	global graph Laplace matrix
M	number of samples
\mathbf{n}_i	position of point cloud \mathcal{N}
N	point cloud with noise
N	number of samples
N_1	number of samples of ground-truth point cloud
N_2	number of samples of observed point cloud after denoising
p_m	block set
p	point of \mathcal{M}
\mathbf{P}_m	block coordinates
P	position matrix of point cloud
\mathbf{P}_x	midpoint coordinate vector in x-direction
\mathbf{P}_y	midpoint coordinate vector in y-direction
\mathbf{P}_z	midpoint coordinate vector in z-direction
q	index of coordinates, $q \in \{x, y, z\}$
r	radius of neighborhood
\mathbb{R}^d	number field with size of d
$\mathbb{R}^{m \times n}$	number field with size of $m \times n$
s	sampling rate of block center
s_1	length of pixel block
s_2	width of pixel block
S	sampling matrix
S_i	average distance
std	multiple threshold of standard deviation
u	image
\mathcal{U}_1	set of ground-truth point cloud
\mathcal{U}_2	set of observed point cloud after denoising
U	ideal model
$w_{m,n}$	edge weight
W	adjacency matrix
x	pixel
$\alpha_i(x)$	coordinate function, $i = 1, \dots, d$
γ	constant
δ	manifold dimension
ε	perturbing noise.
ϵ	reset parameter
ξ	maximum number of iteration
λ	parameter
μ	parameter in penalty item
μ_1	mean value of \mathcal{U}_1
μ_2	mean value of \mathcal{U}_2
σ	standard deviation
σ_1	standard deviation of \mathcal{U}_1
σ_2	standard deviation of \mathcal{U}_2

σ_{12}	covariance of \mathcal{U}_1 and \mathcal{U}_2
τ	edge set
Φ	metric operator
\mathcal{G}	discrete graph
\mathcal{M}	manifold
$ \mathcal{M} $	manifold size
\mathcal{N}	set of point cloud with noise
\mathcal{P}	visible surface block
$\bar{\Omega}$	pixel index

References

- Li, J.; Zhou, Q.; Li, X.; Chen, R.; Ni, K. An improved low-noise processing methodology combined with PCL for industry inspection based on laser line scanner. *Sensors* **2019**, *19*, 3398. [[CrossRef](#)] [[PubMed](#)]
- Dumic, E.; Bjelopera, A.; Nüchter, A. Dynamic Point Cloud Compression Based on Projections, Surface Reconstruction and Video Compression. *Sensors* **2021**, *22*, 197. [[CrossRef](#)]
- Mueller, C.A.; Birk, A. Visual object categorization based on hierarchical shape motifs learned from noisy point cloud decompositions. *J. Intell. Robot. Syst.* **2020**, *97*, 313–338. [[CrossRef](#)]
- Ping, G.; Esfahani, M.A.; Chen, J.; Wang, H. Visual enhancement of single-view 3D point cloud reconstruction. *Comput. Graph. Buildings* **2021**, *12*, 112–119. [[CrossRef](#)]
- Lazzarotto, D.; Ebrahimi, T. Sampling color and geometry point clouds from ShapeNet dataset. *arXiv* **2022**, arXiv:2201.06935.
- Han, S.; Huo, L.; Wang, Y.; Zhou, J.; Li, H. Rapid Reconstruction of 3D Structural Model Based on Interactive Graph Cuts. *Buildings* **2021**, *12*, 22. [[CrossRef](#)]
- Haleem, A.; Gupta, P.; Bahl, S.; Javaid, M.; Kumar, L. 3D scanning of a carburetor body using COMET 3D scanner supported by COLIN 3D software: Issues and solutions. *Mater. Today Proc.* **2021**, *39*, 331–337. [[CrossRef](#)]
- Maru, M.B.; Lee, D.; Tola, K.D.; Park, S. Comparison of depth camera and terrestrial laser scanner in monitoring structural deflections. *Sensors* **2020**, *21*, 201. [[CrossRef](#)] [[PubMed](#)]
- Neuville, R.; Bates, J.S.; Jonard, F. Estimating forest structure from UAV-mounted LiDAR point cloud using machine learning. *Remote Sens.* **2021**, *13*, 352. [[CrossRef](#)]
- Hu, S.; Xiao, S.; Zhang, A.; Deng, Y.; Wang, B. Continuous-Time Laser Frames Associating and Mapping via Multilayer Optimization. *Sensors* **2020**, *21*, 97. [[CrossRef](#)]
- Luo, S.; Hu, W. Differentiable manifold reconstruction for point cloud denoising. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1330–1338.
- Zhou, J.; Jin, W.; Wang, M.; Liu, X.; Li, Z.; Liu, Z. Fast and Accurate Normal Estimation for Point Clouds Via Patch Stitching. *Comput.-Aided Des.* **2022**, *142*, 103121. [[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.* **2019**, *29*, 3474–3489. [[CrossRef](#)]
- Dinesh, C.; Cheung, G.; Bajic, I.V.; Yang, C. Fast 3D point cloud denoising via bipartite graph approximation & total variation. *arXiv* **2018**, arXiv:1804.10831.
- Duan, C.; Chen, S.; Kovačević, J. Weighted multi-projection: 3d point cloud denoising with estimated tangent planes. *arXiv* **2018**, arXiv:1807.00253.
- 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](#)]
- Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 3–15. [[CrossRef](#)]
- Guennebaud, G.; Gross, M. Algebraic point set surfaces. In Proceedings of the ACM SIGGRAPH 2007 Papers, San Diego, CA, USA, 5–9 August 2007; pp. 5–9.
- Öztireli, A.C.; Guennebaud, G.; Gross, M. Feature preserving point set surfaces based on non-linear kernel regression. *Comput. Graph. Forum* **2009**, *28*, 493–501. [[CrossRef](#)]
- Sun, Y.; Schaefer, S.; Wang, W. Denoising point sets via L0 minimization. *Comput. Aided Geom. Des.* **2015**, *35*, 2–15. [[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](#)]
- Lipman, Y.; Cohen-Or, D.; Levin, D.; Tal-Ezer, H. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. (TOG)* **2007**, *26*, 22. [[CrossRef](#)]
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. (TOG)* **2009**, *28*, 1–7. [[CrossRef](#)]
- Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H. Edge-aware point set resampling. *ACM Trans. Graph. (TOG)* **2013**, *32*, 1–12. [[CrossRef](#)]
- Dinesh, C.; Cheung, G.; Bajić, I.V. Point cloud denoising via feature graph laplacian regularization. *IEEE Trans. Image Process.* **2020**, *29*, 4143–4158. [[CrossRef](#)] [[PubMed](#)]

26. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [[CrossRef](#)]
27. Rosman, G.; Dubrovina, A.; Kimmel, R. Patch-Collaborative Spectral Point-Cloud Denoising. *Comput. Graph. Forum* **2013**, *32*, 1–12. [[CrossRef](#)]
28. Deschaud, J.E.; Goulette, F. Point cloud non local denoising using local surface descriptor similarity. *IAPRS* **2010**, *38*, 109–114.
29. Shi, Z.; Osher, S.; Zhu, W. Generalization of the weighted nonlocal laplacian in low dimensional manifold model. *J. Sci. Comput.* **2018**, *75*, 638–656. [[CrossRef](#)]
30. Osher, S.; Shi, Z.; Zhu, W. Low dimensional manifold model for image processing. *SIAM J. Imaging Sci.* **2017**, *10*, 1669–1690. [[CrossRef](#)]
31. Carlsson, G.; Ishkhanov, T.; De Silva, V.; Zomorodian, A. On the local behavior of spaces of natural images. *Int. J. Comput. Vis.* **2008**, *76*, 1–12. [[CrossRef](#)]
32. Shuman, D.I.; Narang, S.K.; Frossard, 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](#)]
33. Zeng, J.; Pang, J.; Sun, W.; Cheung, G. Deep graph Laplacian regularization for robust denoising of real images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 15–20 June 2019.
34. Pang, J.; Cheung, G. Graph Laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Trans. Image Process.* **2017**, *26*, 1770–1785. [[CrossRef](#)] [[PubMed](#)]
35. Hein, M. Uniform convergence of adaptive graph-based regularization. In Proceedings of the International Conference on Computational Learning Theory, Pittsburgh, PA, USA, 22–25 June 2006; pp. 50–64.
36. Liu, X.; Cheung, G.; Wu, X. Random walk graph Laplacian-based smoothness prior for soft decoding of JPEG images. *IEEE Trans. Image Process.* **2016**, *26*, 509–524. [[CrossRef](#)] [[PubMed](#)]
37. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]