Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6432929, 14 pages
https://doi.org/10.1155/2021/6432929

Research Article

Dataset Denoising Based on Manifold Assumption

Zhonghua Hao,[1,2] **Shiwei Ma**,[3] **Hui Chen**,[4] **and Jingjing Liu**[5]

[1]*Qingdao University, College of Automation, Qingdao 266071, China*
[2]*Qingdao University, College of Electrical Engineering, Qingdao 266071, China*
[3]*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200444, China*
[4]*College of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China*
[5]*State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201210, China*

Correspondence should be addressed to Shiwei Ma; masw@shu.edu.cn

Received 25 May 2020; Revised 1 November 2020; Accepted 22 December 2020; Published 18 January 2021

Academic Editor: Jun Shen

Learning the knowledge hidden in the manifold-geometric distribution of the dataset is essential for many machine learning algorithms. However, geometric distribution is usually corrupted by noise, especially in the high-dimensional dataset. In this paper, we propose a denoising method to capture the "true" geometric structure of a high-dimensional nonrigid point cloud dataset by a variational approach. Firstly, we improve the Tikhonov model by adding a local structure term to make variational diffusion on the tangent space of the manifold. Then, we define the discrete Laplacian operator by graph theory and get an optimal solution by the Euler–Lagrange equation. Experiments show that our method could remove noise effectively on both synthetic scatter point cloud dataset and real image dataset. Furthermore, as a preprocessing step, our method could improve the robustness of manifold learning and increase the accuracy rate in the classification problem.

1. Introduction

Since objects vary gradually in the real world, the manifold assumption indicates that the data points depict the state of an object should distribute on a smooth low-dimensional manifold embedded in high-dimensional observation space [1]. Dimensionalities of the manifold are key factors that control variation of the object state. For example, in Figure 1, the images of the rotational duck toy distribute on a one-dimensional manifold (a curve) embedded in high-dimensional pixel space. Each image depicts a particular state of the duck. Although the pixel values change dramatically at these images, humans could discover easily that they are controlled by one key factor: rotation of the duck.

Learning the knowledge hidden in the manifold-geometric distribution of a high-dimensional dataset is essential in many machine learning algorithms. For example, manifold learning algorithms aim to discover the nonlinear geometric structure dataset by preserving different local geometric properties [3–8]. The embedding results can be further used in data visualization, motion analysis, and classification [9, 10]. Moreover, much research takes manifold assumption as a constraint condition in its objective function [11, 12]. It is worth noting that manifold assumption is applied to explain why deep learning works well recently [13–15]. This research indicates deep learning could capture the manifold structure of one kind of knowledge by powerful nonlinear mapping.

However, noise is inevitable in data acquisition. For example, in Figure 1, the noiseless images of the rotational duck toy (red points) should lie on a curve embedded in the pixel space. However, due to the long exposure time and camera shake, the duck becomes "brighten" and "small" in the image. The corresponding noise data point, which is marked by "N" and green color in Figure 1, does not lie on the curve because pixel values change dramatically in the noise image.

Noise makes machine learning models fragile and hard to train. For example, the outlier points are difficult to handle in the classification and clustering task. Machine learning model needs to become more complex to get proper results [13]. In manifold learning algorithms, noise points make recovered embeddings difficult to capture the true

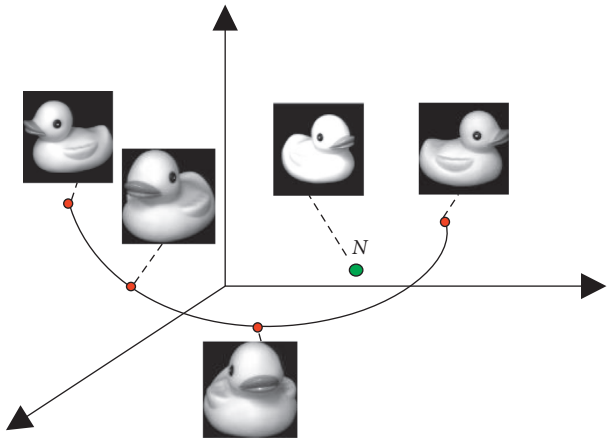

Figure 1: Image dataset of the rotational duck toy distributes on a one-dimensional manifold [2]. The red points correspond to noiseless images. The green point correspond to the noise image.

manifold-geometric distribution of the dataset. The reason is that the "short circuit" phenomenon arises easily in the noise dataset which destroys the local linear structure of the manifold [16].

In this paper, we propose a novel denoising method based on manifold assumption. Our aim is to obtain the data points that lie on the noiseless manifold through noise data points. Compared with the existing denoising methods, our method has two contributions worth being highlighted:

(1) Our method makes use of manifold-geometric distribution information of the dataset. Therefore, this method works for a dataset rather than a single data point.

(2) Our method improves the Tikhonov model to make the variational diffusion on the tangent space of the manifold for a high-dimensional nonrigid point cloud dataset.

Our method could capture the "true" geometric structure of the noise dataset. After denoising, the key factors that control the geometric distribution of the dataset are maintained and the characteristics of individual points are removed as noise. As a preprocessing step, our method could improve the robustness of manifold learning and increase the accuracy rate in the classification problem.

The rest of the paper is organized as follows: a brief review of the research on the manifold assumption is outlined in Section 1. Section 2 describes the motivation and details of the proposed method. In Section 3, experiments are conducted on both synthetic and real data to evaluate our method. Section 4 concludes remarks and a discussion of future work.

2. Related Work

Existing denoising methods always work for the noise in a single data point, such as "Gaussian noise" or "pepper noise" [17, 18] in an image. However, these methods could not deal with the noise that distorts the geometric distribution of the dataset, such as the noise duck toy image (green point) caused by longer exposure time and camera shake in Figure 1.

Only a few studies exist to deal with this problem. Gong et al. [19] proposed a local linear denoising method. This method removed noise by projecting noise data points to the tangent space of manifold which is estimated by the principal component analysis method firstly. Then, local denoised patches are aligned to get the global denoising dataset. However, the principal components may be distorted because they are calculated by the neighborhood of noise data points, which could lead to a wrong denoising result. Hao et al. [16] also utilized principal component analysis and projection method to find the noiseless data points. Therefore, it has the same problem. Moreover, many machine learning methods proposed the noise-resistant model for outliers but did not discuss denoising as an independent problem [7, 20]. For example, Zhang et al. [7] proposed an adaptive neighborhood selection method by the shrink and expand strategy to resist noise on the neighborhood of manifold.

In this paper, we propose a denoising method for the dataset. This method improves the Tikhonov method by adding a local structure term. The optimal solution is obtained by minimizing the objective function through a variational diffusion approach.

3. Proposed Approach

Let $\mathbf{F} = \{\mathbf{f}(1), \mathbf{f}(2), \ldots, \mathbf{f}(\mathbf{m})\}$ be the noise dataset. $\mathbf{f}(\mathbf{x}) \in R^D$ is the x-th data point in F. D is the dimension number of $\mathbf{f}(\mathbf{x})$. Let $\mathbf{U} = \{\mathbf{u}(1), \mathbf{u}(2), \ldots, \mathbf{u}(\mathbf{m})\}$ be the noiseless dataset we want to obtain. $\mathbf{u}(\mathbf{x}) \in R^D$ is the x-th data point in U. $\mathbf{f}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \xi(\mathbf{x}), \xi(\mathbf{x}) \in R^D$ is the noise of $\mathbf{f}(\mathbf{x})$. The goal is to recover U from F.

We illustrate our method in three steps: firstly, introduce to inspiration and motivation; then, construct the objective function by improving the Tikhonov model; and finally, optimize the objective function and get the solution by taking discrete operators.

3.1. Inspiration and Motivation. Manifold assumption claims that the noiseless data point $\mathbf{u}(\mathbf{x})$ that depicts the object state (the blue points in Figure 2) should lie on a smooth manifold $\mathcal{U}$ (blue surface in Figure 2) embedded in observation space. However, noise points $\mathbf{f}(\mathbf{x})$ (red points) distribute on the noise manifold $\mathcal{F}$. The denoising problem is how to obtain ux on $\mathcal{U}$from $\mathbf{f}(\mathbf{x})$ on $\mathcal{F}$.

3.2. Objective Function. The objective function is formulated in this part. Firstly, we illustrate the Tikhonov model briefly in image denoising which is similar to our problem. Then, the challenge of our problem is shown. Finally, we improve the Tikhonov model and construct the objective function for our problem.

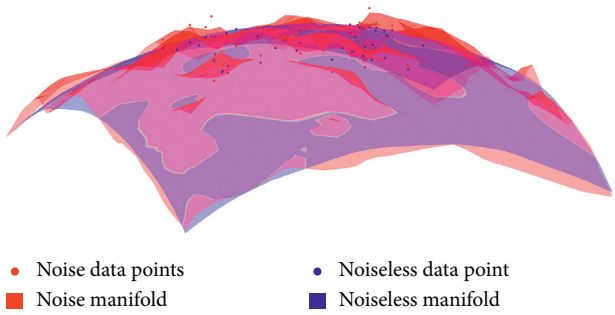

Figure 2: Illustration of the idea of our method: obtain the noiseless blue points that lie on smooth manifold (blue surface) from the noise red points that distribute on an irregular surface (noise manifold).

3.2.1. Tikhonov Model in the Image Denoising Problem. Our problem is similar to the image denoising problem $f(x, y) = u(x, y) + \xi(x, y)$, where x and y are row and column numbers of a pixel in an image. $f(x, y)$ and $u(x, y)$ are pixel values at row $x$and column y in noise and noiseless image, respectively. $\xi(x, y)$ is the noise. In Figure 2, if we regard the x, y, and z coordinate of $\mathbf{f}(\mathbf{x})$ as row number, column number, and pixel value, then the red manifold F depicts the pattern of noise image. Therefore, the image denoising problem is to find a noiseless image $\mathcal{U}$ from $\mathcal{F}$.

The Tikhonov model is one of the most classical variational models to deal with this problem [21]:

$$E(u) = \min_u \frac{1}{2}\int_{|\Omega} (u - f)^2 \mathrm{d}x + \frac{\alpha}{2}\int_{|\Omega} |\nabla \mathbf{u}|^2 \mathrm{d}x, \quad (1)$$

where Ω is the image domain and dx is the area element (pixel) in Ω. ∇u is the gradient of $u(x)$. The first term $\int_\Omega (u - f)^2$ is "data term" that measures the Euclidean distance between $\mathcal{F}$ and $\mathcal{U}$. The second term $\int_\Omega |\nabla \mathbf{u}|^2 \mathrm{d}x$ is "smooth term" that measures the noise strength of $\mathcal{U}$. Since these two terms have opposite effect, the parameter α balances these two terms. If α is small, $\mathcal{U}$ is close to $\mathcal{F}$ but the noise strength is large. On the other hand, the noise becomes small but the image pattern of $\mathcal{U}$ is "unlike" $\mathcal{F}$.

3.2.2. The Challenge of Our Problem. In the image denoising problem, the gradient operator is defined as [21]

$$\nabla \mathbf{u} = [u(x, y) - u(x - 1, y), u(x, y) - u(x, y - 1)]^T. \quad (2)$$

When minimizing the "smooth term" $\int_\Omega |\nabla \mathbf{u}|^2 \mathrm{d}x$ in (1), the pixel values in the image became the same, whereas the image area does not change since x and y are fixed.

However, in our problem, the dataset is nonrigid and high-dimensional cloud points. Let $\mathbf{u}(\mathbf{x}) = [u(x)^1, u(x)^2, \ldots, u(x)^D] \in R^D$ be a data point. D is the dimension number of $\mathbf{u}(\mathbf{x})$. Suppose $\mathcal{N}_{u(x)}$ is the neighborhood of $\mathbf{u}(\mathbf{x})$ which is determined by the KNN method:

$$\mathcal{N}_{u(x)} = \{\mathbf{u}(y_i) \in \mathcal{N}_{u(x)}\}, \quad i = 1, \ldots, k. \quad (3)$$

Naturally, the gradient operator is defined as

$$\nabla \mathbf{u} = [\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y}_1), \mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y}_2), \ldots, \mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y}_\mathbf{k})]^T. \quad (4)$$

Therefore, the "smooth term" in (1) is

$$\int_\Omega |\nabla \mathbf{u}|^2 = \int_\Omega \int_{\mathcal{N}_{u(x)}} (u(x) - u(y_i))^2 \mathrm{d}y\, \mathrm{d}x. \quad (5)$$

When minimizing an objective function, the "smooth term" makes $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{y_i})$ become the same point. Therefore, the "cluster" phenomenon arises in the dataset—some points are brought close together and the other points are pushed away. Therefore, the geometric structure of the manifold $\mathcal{U}$ (blue surface in Figure 2) will shrink to a few point clusters rather than becoming smooth. Therefore, the Tikhonov model could not be applied directly to solve our problem.

3.2.3. Our Objective Function. To deal with this problem, we maintain the geometric distribution of $\mathcal{U}$ by keeping the tangent linear structure when minimizing the objective function. Since the neighborhood of the manifold could be regarded as tangent space (the blue plane in Figure 3), we make the neighborhood structure of $\mathcal{U}$ the same as $\mathcal{F}$.

The weight of local linear representation is utilized to depict the geometric structure of the neighborhood. The weight $\mathbf{W_f}$ of data point $\mathbf{f}(\mathbf{x})$ is defined as

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{k} W_{fi} \mathbf{f}(\mathbf{y_i}), \quad (6)$$

where $\mathbf{f}(\mathbf{y_i}) \in \mathcal{N}_{f(x)}$ and W_{fi} is the i-th component of $\mathbf{W_f}$ between $\mathbf{f}(\mathbf{x})$ and $\mathbf{f}(\mathbf{y_i})$. Similarly, the linear representation weight of $\mathbf{u}(\mathbf{x})$ is defined as $\mathbf{W_u}$.

The local linear structure can be maintained if we set $\mathbf{W_u}$ the same as $\mathbf{W_f}$. Then, $\mathbf{f}(\mathbf{x})$ could only move along the normal space of manifold when minimizing the "smooth term" in the objective function because the tangent geometric structure is fixed by $\mathbf{W_u}$. Therefore, we add a "local structure term" in the Tikhonov model:

$$\int_\Omega (\mathbf{u}(\mathbf{x}) - \int_{\mathcal{N}_{u(x)}} W_{fi} \mathbf{f}(\mathbf{y_i}) \mathbf{dy})^2 \mathbf{dx}, \quad (7)$$

where $\int_{\mathcal{N}_{u(x)}} W_{fi} \mathbf{f}(\mathbf{y_i}) \mathbf{dy}$ is the linear reconstruction of $\mathbf{u}(\mathbf{x})$. Thus, our objective function is

$$E(\mathbf{u}) = \min \frac{1}{2}\int_\Omega (\mathbf{u} - \mathbf{f})^2 \mathbf{dx} + \frac{\alpha}{2}\int_\Omega |\nabla \mathbf{u}|^2 \mathbf{dx} + \frac{\beta}{2}\int_\Omega (\mathbf{u} - \mathbf{W}\mathbf{f}_{\mathcal{N}})^2 \mathbf{dx}, \quad (8)$$

where α and β are balance parameters.

3.3. Optimal Solution. In this part, we get optimal $\mathbf{u}$ by minimizing objective function (8). The solution in the continuous form is calculated firstly. Then, the discrete operator is defined and plugged to get a discrete solution.

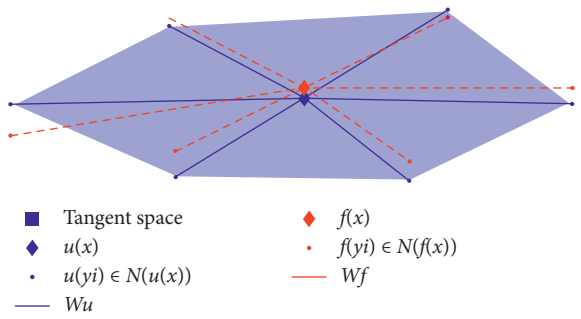

FIGURE 3: Local weights in the neighborhood. $\mathbf{W_u}$ is the weight of the linear representation of $\mathbf{u}(\mathbf{x})$ by $\mathbf{u}(\mathbf{y_i})$ that are in the neighborhood of $\mathbf{u}(\mathbf{x})$. $\mathbf{W_f}$ is the weight of the linear representation of $\mathbf{f}(\mathbf{x})$ by $\mathbf{f}(\mathbf{y_i})$ that are in the neighborhood of $\mathbf{f}(\mathbf{x})$.

3.3.1. Solution in Continuous Form. To get optimal $\mathbf{u}$, we calculate the derivative of (8) with respect to $\mathbf{u}$ by variational approach and set it to zero:

$$\begin{aligned} E'(\varepsilon) &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon} E(\mathbf{u}+\varepsilon\boldsymbol{\eta}, \nabla\mathbf{u}+\varepsilon\nabla\mathbf{u}) \\ &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\frac{1}{2}\int_\Omega (\mathbf{u}+\varepsilon\boldsymbol{\eta}-\mathbf{f})^2\mathbf{dx} + \frac{\alpha}{2}\int_\Omega |\nabla\mathbf{u}+\varepsilon\nabla\boldsymbol{\eta}|^2\mathbf{dx} + \frac{\beta}{2}\int_\Omega (\mathbf{u}+\varepsilon\boldsymbol{\eta}-\mathbf{W}\mathbf{f}_{\mathcal{N}})^2\mathbf{dx}\right) \\ &= \int_\Omega \boldsymbol{\eta}(\mathbf{u}-\mathbf{f})\mathbf{dx} + \alpha\int_\Omega \nabla\boldsymbol{\eta}\nabla\mathbf{u}\,\mathbf{dx} + \beta\int_\Omega \boldsymbol{\eta}(\mathbf{u}-\mathbf{W}\mathbf{f}_{\mathcal{N}})\mathbf{dx} \\ &= \int_\Omega \boldsymbol{\eta}(\mathbf{u}-\mathbf{f})\mathbf{dx} + \alpha\int_{\partial\Omega} \vec{n}\boldsymbol{\eta}\nabla\mathbf{u}\,\mathbf{ds} - \alpha\int_\Omega \boldsymbol{\eta}\Delta\mathbf{u}\,\mathbf{dx} + \beta\int_\Omega \boldsymbol{\eta}(\mathbf{u}-\mathbf{W}\mathbf{f}_{\mathcal{N}})\mathbf{dx} \\ &= \eta\int_\Omega [\,(\mathbf{u}-\mathbf{f}) - \alpha\Delta\mathbf{u} + \beta(\mathbf{u}-\mathbf{W}\mathbf{f}_{\mathcal{N}})\,]\mathbf{dx} + \alpha\int_{\partial\Omega} \vec{n}\boldsymbol{\eta}\nabla\mathbf{u}\,\mathbf{ds}. \end{aligned} \tag{9}$$

Therefore, the Euler–Lagrange equation of $\mathbf{u}$ is

$$(\mathbf{u}-\mathbf{f}) - \alpha\Delta\mathbf{u} + \beta(\mathbf{u}-\mathbf{W}\mathbf{f}_{\mathcal{N}}) = 0. \tag{10}$$

Then,

$$\mathbf{u} = \frac{\mathbf{f}+\beta\mathbf{W}\mathbf{f}_{\mathcal{N}}+\alpha\Delta\mathbf{u}}{1+\beta}. \tag{11}$$

And the boundary condition is

$$\vec{n}\nabla\mathbf{u} = 0. \tag{12}$$

3.3.2. Solution in Discrete Form. To get the discrete solution, we define the discrete Laplacian operator in (11) by spectral graph theory [22]. Firstly, the gradient of $\mathbf{u}(\mathbf{x})$ is defined as

$$\nabla_{\mathbf{wG}}\mathbf{u}(\mathbf{x},\mathbf{y}) = \{(\mathbf{u}(\mathbf{y_i}) - \mathbf{u}(\mathbf{x}))\mathbf{W_d}(\mathbf{x},\mathbf{y})\}_{\mathbf{u}(\mathbf{y_i})\in\mathcal{N}_{u(x)}}, \quad i = 1,\ldots,k. \tag{13}$$

This gradient is a k-dimensional vector because there are k data points in $\mathcal{N}_{u(x)}$. The subscript "**wG**" is abbreviated to "weighted graph." $\mathbf{W_d}(\mathbf{x},\mathbf{y})$ is a weight vector. The component $\mathbf{W_d}(\mathbf{x},\mathbf{y_i})$ should be important if $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{y_i})$ are near. On the contrary, the component should be unimportant if $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{y_i})$ are far away. Therefore, we define $\mathbf{W_d}(\mathbf{x},\mathbf{y})$ as

$$\mathbf{W_d}(\mathbf{x},\mathbf{y}) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{\mathbf{d}(\mathbf{x},\mathbf{y})}{2\sigma^2}\right), \tag{14}$$

where $\mathbf{d}(\mathbf{x},\mathbf{y})$ is the vector of Euclidean distance between $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{y_i}) \in \mathcal{N}_{u(x)}$. σ is the variance of $\mathbf{d}(\mathbf{x},\mathbf{y})$. For the convenience of calculations, we set $\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})} = \mathbf{W_d}(\mathbf{x},\mathbf{y})$. Therefore, the discrete gradient of $\mathbf{u}(\mathbf{x})$ is

$$\nabla_{\mathbf{wG}}\mathbf{u}(\mathbf{x},\mathbf{y}) = \left\{(\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x}))\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\right\}_{\mathbf{u}(\mathbf{y})\in\mathcal{N}_{u(x)}}. \tag{15}$$

Consequently, the gradient of a vector $\mathbf{v}(\mathbf{x},\mathbf{y})$ is (the derivation procedure is listed at "Notice" at the end of this capture):

$$\nabla_{\mathbf{wG}}\mathbf{v} = -\sum_{\mathbf{y}\in\mathcal{N}(x)} (\mathbf{v}(\mathbf{y},\mathbf{x}) - \mathbf{v}(\mathbf{x},\mathbf{y}))\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}. \tag{16}$$

Let $\mathbf{v}(\mathbf{x},\mathbf{y}) = \nabla_{\mathbf{wG}}\mathbf{u}(\mathbf{x},\mathbf{y}) = (\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x}))\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}$, therefore, the discrete Laplace operator of $\mathbf{u}(\mathbf{x})$ can be defined by

$$\begin{aligned}\Delta_{\mathbf{wG}}\mathbf{u} = \nabla_{\mathbf{wG}}\left(\nabla_{\mathbf{wG}}\mathbf{u}\right) &= \sum_{y\in\mathcal{N}(x)}\left(\left(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x})\right)\right.\\ &\left.\cdot\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}-\left(\mathbf{u}(\mathbf{x})-\mathbf{u}(\mathbf{y})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\right)\\ &= 2\sum_{y\in\mathcal{N}(x)}\left(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x})\right)\mathbf{w}(\mathbf{x},\mathbf{y}).\end{aligned} \tag{17}$$

We plug the discrete Laplace operator into (11). The solution of our object energy function (8) is

$$\mathbf{u}(\mathbf{x})^{k+1} = \frac{\mathbf{f}(\mathbf{x})+\beta\sum_{y\in\mathcal{N}(x)}\mathbf{W}(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})+2\alpha\sum_{y\in\mathcal{N}(x)}\left(\mathbf{u}(\mathbf{y})^{k}-\mathbf{u}(\mathbf{x})^{k}\right)\mathbf{w}(\mathbf{x},\mathbf{y})}{1+\beta}, \tag{18}$$

where the superscripts k and k+1 are the iteration step. The initial value of $\mathbf{u}$ is set to $\mathbf{f}$. The optimal $\mathbf{u}$ is obtained by iteration, which ends up when $E(\mathbf{u}) < \varepsilon$, where $E(\mathbf{u})$ is the objective function value and ε is a small error we set. The boundary condition (12) could be ignored because the dataset is scattered and nonrigid cloud points.

Notice:

The gradient of a vector $\mathbf{v}$ could be derived as follows:

$$\begin{aligned}\sum_{x\in\Omega}\nabla_{\mathbf{wG}}\mathbf{u}\cdot\mathbf{v} &= \sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\left(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\mathbf{v}(\mathbf{x},\mathbf{y})\\ &= \frac{1}{2}\sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\left(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\mathbf{v}(\mathbf{x},\mathbf{y})\\ &\quad+\sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\left(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\mathbf{v}(\mathbf{x},\mathbf{y})\\ &= \frac{1}{2}\sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\mathbf{u}(\mathbf{x})\left(\mathbf{v}(\mathbf{y},\mathbf{x})-\mathbf{v}(\mathbf{x},\mathbf{y})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\\ &\quad+\frac{1}{2}\sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\mathbf{u}(\mathbf{y})\left(\mathbf{v}(\mathbf{x},\mathbf{y})-\mathbf{v}(\mathbf{y},\mathbf{x})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\\ &= \sum_{x\in\Omega}\sum_{y\in\mathcal{N}(x)}\left(\mathbf{v}(\mathbf{y},\mathbf{x})-\mathbf{v}(\mathbf{x},\mathbf{y})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}\mathbf{u}(\mathbf{x})\\ &= -\sum_{x\in\Omega}\nabla_{\mathbf{wG}}\mathbf{v}\cdot\mathbf{u}.\end{aligned} \tag{19}$$

Therefore,

$$\nabla_{wG}\mathbf{v} = -\sum_{y\in\mathcal{N}(x)}\left(\mathbf{v}(\mathbf{y},\mathbf{x})-\mathbf{v}(\mathbf{x},\mathbf{y})\right)\sqrt{\mathbf{w}(\mathbf{x},\mathbf{y})}. \tag{20}$$

4. Experiments

In this section, we evaluate our algorithm on both the synthetic scatter point cloud dataset and real image dataset. Then, this method is utilized as a preprocess step for manifold learning and classification task. The major parameters of our algorithm include (1) the neighborhood size k; (2) the smooth term weight α; and (3) the local structure term weight β.

4.1. Experiments on Synthetic 3D Scatter Cloud Data. In this part, we test our algorithm on the classical "swiss roll" dataset. The data points are sampled from 2D manifold randomly embedded in the 3D space like a swiss roll cake. Figures 4(a) and 4(b) at first row are noiseless and noise dataset at [−8, 10] and [0, 0] viewpoint, respectively. It is obvious that noise data points distribute around the "swiss roll" manifold but do not lie on it exactly. Our goal is to recover the noiseless dataset in Figure 4(a) by the noise dataset in Figure 4(b). In this experiment, we set the number of data points $n = 1300$, KNN parameter $k = 12$, and the noise parameter NI = 1. The MATLAB code of the swiss roll dataset is listed in Table 1.

The second, third, and fourth rows in Figure 4 are denoising results by our method with α and β equal to (1, 1), (3, 1), and (0.3, 1), respectively. For ease of viewing, we set the denoising datasets at [−8, 10] and [0, 0] viewpoints in the left and right columns. In the right column, it is easy to see that the denoising data points are closed to the tangent space of manifold compared with (b), which show that our method is effective. Among them, (f) seems to be the best result because the denoising points are the nearest to manifold compared with (d) and (h). However, the "cluster" phenomenon arises in the denoising dataset; some points are close together and the other points are pushed away, which is easy to see in (e). The reason is that the large smooth parameter ($\alpha = 3$) makes geometric distribution distort when minimizing the objective function. Conversely, the "cluster" phenomenon in (g) is not serious when we set a small parameter $\alpha = 0.3$, but the noise is large.

To conduct a quantitative comparison between noise and denoising datasets, we assess the quality of the denoising datasets by mean square error (MSE) and tangent distance error (TE). MSE is a widely used index which measures the average squared Euclidean distance difference between two datasets:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(u_i - u_i^*\right)^2, \tag{21}$$

where N is the point number of the dataset. u_i and u_i^* are a noise data point and corresponding noiseless data point.

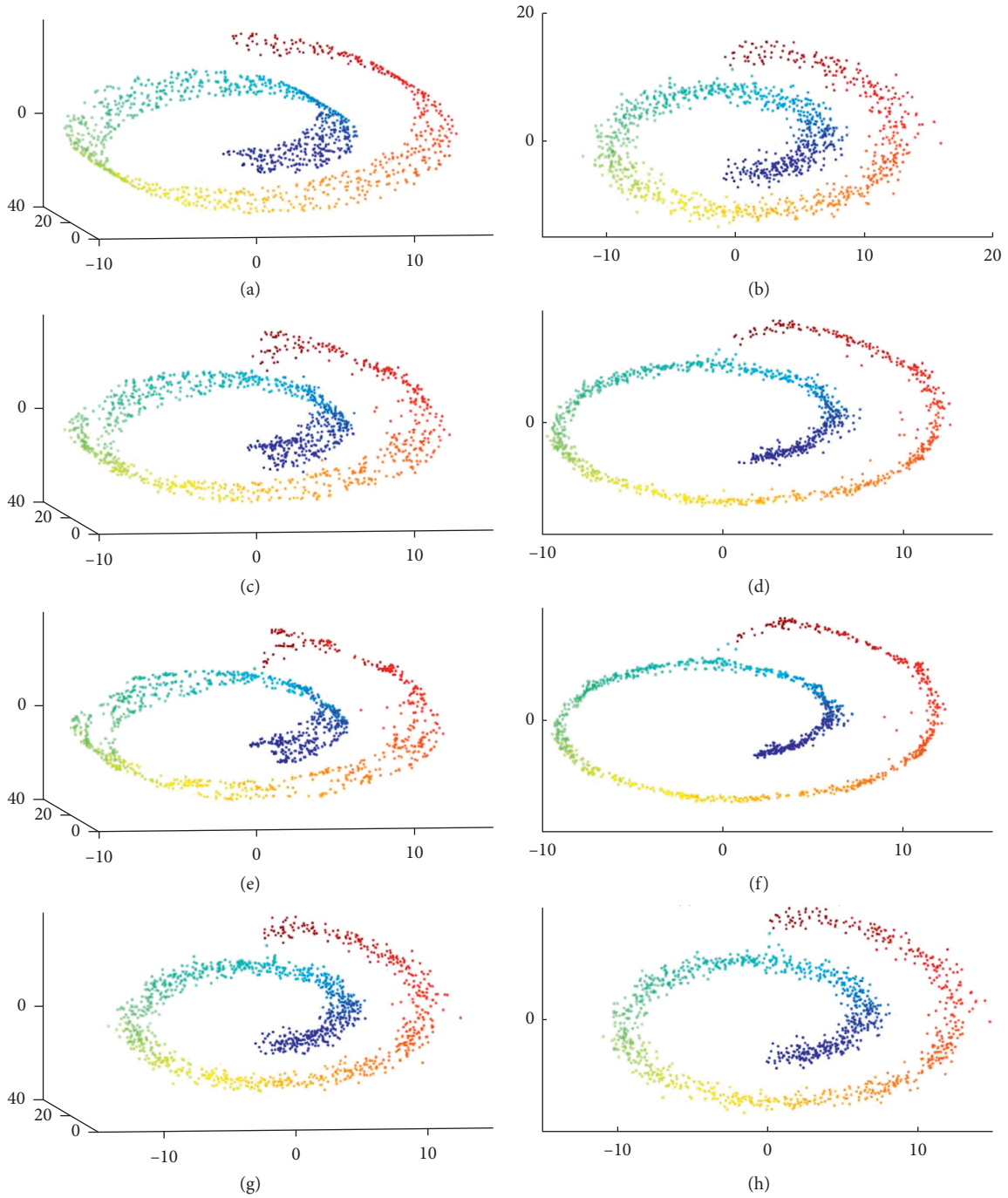

Figure 4: Denoising results with different parameters. (a) Noiseless dataset. (b) Noise dataset. (c) and (d) Denoising results with $\alpha = 1$ and $\beta = 1$. (e) and (f) Denoising results with $\alpha = 3$ and $\beta = 1$. Large α eliminates noise, but "cluster" phenomenon arises. (g) and (h) Denoising results with $\alpha = 0.3$ and $\beta = 1$.

Table 1: MATLAB code for the noise swiss roll dataset.

```
Input: the number of datasets: n; noise parameter: NI
Output: swiss roll dataset, noiseless and noise
t = (3*pi/2)*(1 + 2*rand(n, 1));
Height = 30*rand (n, 1);
Noiseless data = [t .*cos(t) height t .*sin(t)];
Noise data = [t .*cos(t) height t .*sin(t)] + NI*randn(n, 3);
```

The tangent distance error (TE) measures the distance of u_i to the tangent space of the manifold. A small TE indicates that u_i lies on the manifold and noise is weak. On the contrary, the noise strength is large if TE is big. For the convenience of calculations, we approximate TE as the Euclidean distance between u_i and its nearest data point in the noiseless dataset. The tangent distance error (TE) is defined as

$$\mathrm{TE} = \frac{1}{N}\sqrt{\sum_{i=1}^{N} u_i - \min_{u_i^*} d\left(u_i, u_i^*\right)^2}, \quad \text{s.t. } u_i^* \in U^*, \tag{22}$$

where N is the number of data points, u_i and u_i^* represent the denoising data point and noiseless data point, respectively. U^* is the noiseless dataset.

To evaluate our algorithm, we test seven sets of α and β ranging from 0 to 10. MSE and TE are listed in Tables 2 and 3. When α and β equal 0, the "data term" is the only term remaining in the objective function (8). Therefore, the denoising dataset is the same as the noise dataset and the value at $(\alpha = 0, \beta = 0)$ is the errors of the noise dataset. While α is small and β is large, the "data term" and "local structure term" maintain the geometric structure of the noise dataset. Therefore, the errors at the upper right of the table are close to the errors of the noise dataset. While α is large and β is small, the "smooth term" plays a major role. It could lead to a "cluster" phenomenon which distorts the geometric structure of the dataset and make errors large at the bottom left of the table. It is able to see that the errors near the diagonal of tables are much smaller than the others.

Table 2: MSE of our method (10^{-1}).

α β	0	0.2	0.5	0.8	1	3	10
0	2.53	2.53	2.53	2.53	2.53	2.53	2.53
0.2	2.36	2.30	2.25	2.27	2.30	2.35	2.49
0.5	3.00	2.67	2.44	2.34	2.33	2.28	2.40
0.8	3.77	3.18	2.76	2.55	2.46	2.26	2.34
1	4.28	3.52	3.00	2.73	2.60	2.30	2.33
3	9.63	7.29	5.54	4.67	4.22	2.82	2.30
10	28.4	21.5	15.3	11.9	10.6	5.37	2.91

Table 3: ET of our method (10^{-2}).

α β	0	0.2	0.5	0.8	1	3	10
0	2.01	2.01	2.01	2.01	2.01	2.01	2.01
0.2	1.70	1.73	1.74	1.80	1.81	1.91	1.95
0.5	1.68	1.67	1.69	1.70	1.70	1.80	1.91
0.8	1.74	1.69	1.66	1.67	1.68	1.72	1.89
1	1.80	1.72	1.69	1.65	1.66	1.72	1.87
3	2.42	2.12	1.93	1.82	1.80	1.66	1.71
10	4.55	3.73	3.14	2.60	2.47	1.89	1.65

4.2. Experiments on the Image Dataset. In this part, we test our method on two real image datasets: MNIST handwritten number dataset [23] and "LLE face" dataset. Image is regarded as a point in pixel space. For example, the image in the MNIST dataset could be regarded as a point in 784-dimensional space because it has 784 pixels. Therefore, the only difference between this part to experiment 3.1 is that the dimensionality of image-point is much higher than the synthetic scatter point in 3D space.

We analyze denoising images both from the subjective and objective aspects. Firstly, our method is applied to raw image datasets. Ideally, key factors that control the geometric distribution of the dataset could be maintained and the characteristics in individual images are removed as noise. Since there is no ground truth of the raw image dataset, we could only evaluate results by eyes subjectively. Secondly, we add several types of noise in an image and utilize MSE to measure the denoising images by our method and classical image denoising methods objectively.

4.2.1. Experiments on the Raw Image Dataset. We select "number 3" and "number 4" datasets in MNIST which contain 1010 and 982 images, respectively. The size of each image is $28 * 28$ pixels. The "LLE face" dataset contains 1965 face images with different expressions and shooting angles. The size of each image is $28 * 20$ pixels.

Figure 5 shows 110 images in the "handwritten number 3" dataset. The left side is original images and the right side is the corresponding denoising images by our method. In this experiment, $k = 15$, $\alpha = 0.8$, and $\beta = 1$. Four typical images are marked with a box and listed in Figure 5. It can be seen that the blurring strokes become clear and the posture of number in the image is maintained.

Figure 6 shows the 110 images in the "handwritten number 4" dataset. The left and right sides are original images and the corresponding denoising images by our method, respectively. In this experiment, $k = 15$, $\alpha = 8$, and $\beta = 1$. It can be seen that the denoising images maintain the main factors, such as the angularity of number "4." And the individual characteristics are removed after denoising; for example, the difference of stroke width becomes small after denoising. Four typical images are marked with a box and listed in Figure 6. It is obvious that the margin of "head" of number "4" becomes large in the first two images after denoising. In the third image, the stroke width becomes broad. In the fourth image, the "bend" at the upside of the stroke is removed.

Figure 7 shows the denoising result for the LLE face dataset. This dataset contains 1965 face images and the size of each image is $28 * 20$ pixels. In this experiment, $k = 15$, $\alpha = 3$, and $\beta = 0.8$. [4] shows that this dataset distributes on the manifold that spans by two key factors: head pose and expression, where the expression reflects by lip shape in images.

It can be seen that these two factors are maintained after denoising and the characters in the individual image are removed as noise. Four typical images are marked with a box and listed in Figure 7. In the first two images, the head twists to the left and right slightly in the original dataset whereas the head pose is fixed after denoising. In the third image, the original head seems to be smaller than the other images which may be caused by camera shake. The corresponding denoising image enlarges the face, and the cheek and chin became "fat." In the fourth image, the eyes are "open" after denoising.

4.2.2. Experiments on the Noise Image. In this part, we add several different types of noise to an LLE face image. Then, our method and three classical image denoising methods are

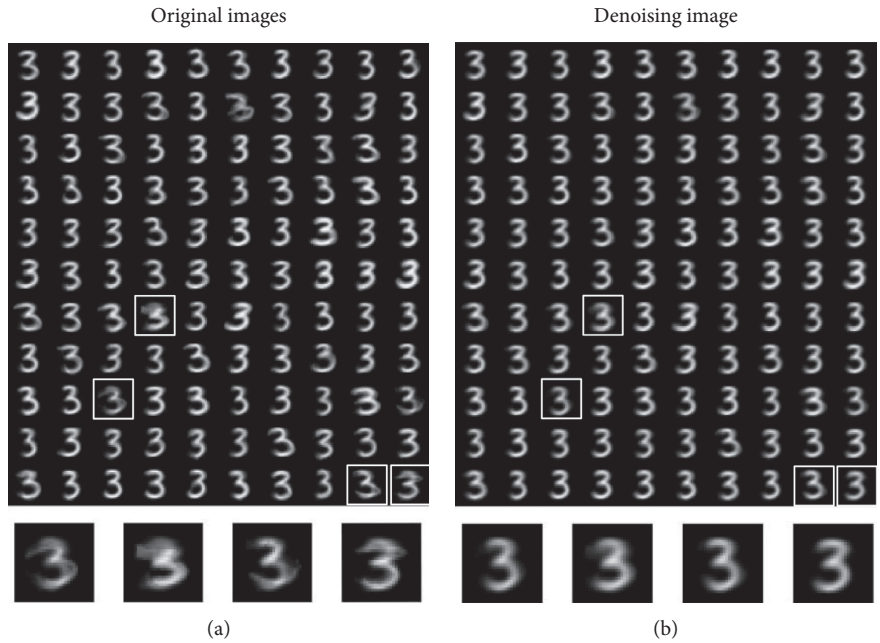

Figure 5: Denoising for the MNIST number 3 dataset. Original images and corresponding denoising images are listed in the left column and right column. The blurring strokes become clear.

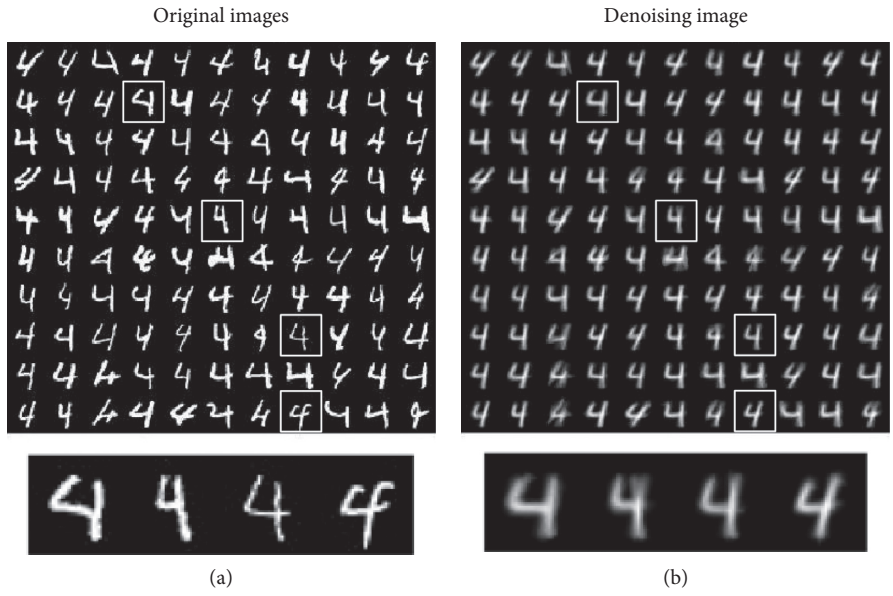

Figure 6: Denoising for the MNIST number 4 dataset. Original images and corresponding denoising images are listed in the left column and right column. The stroke widths become similar, and the posture of number 4 is maintained.

applied to these noise images. Finally, MSE is utilized to evaluate denoising images.

Figure 8 shows the denoising images by four denoising methods for five types of noise. The first column is a raw LLE face image. Brightness noise, Gaussian noise, salt and pepper noise, rotation noise, and scaling noise are added to the raw image which are shown in the second column, top to bottom row. The MATLAB code of noise model is listed in Table 4.

Three classical denoising methods, mean filtering, median filtering, and Tikhonov method are utilized to deal with these noise images. The corresponding denoising images are listed in the third, fourth, and fifth columns in Figure 8. The images in the last column are denoising results by our method. MSE is listed below each image. In this experiment, the size of the raw LLE face image is $28 * 20$ pixels. In mean filtering, the size of the filter is $2 * 2$ pixels. In median filtering, the size of the filter is $3 * 3$ pixels. In the Tikhonov

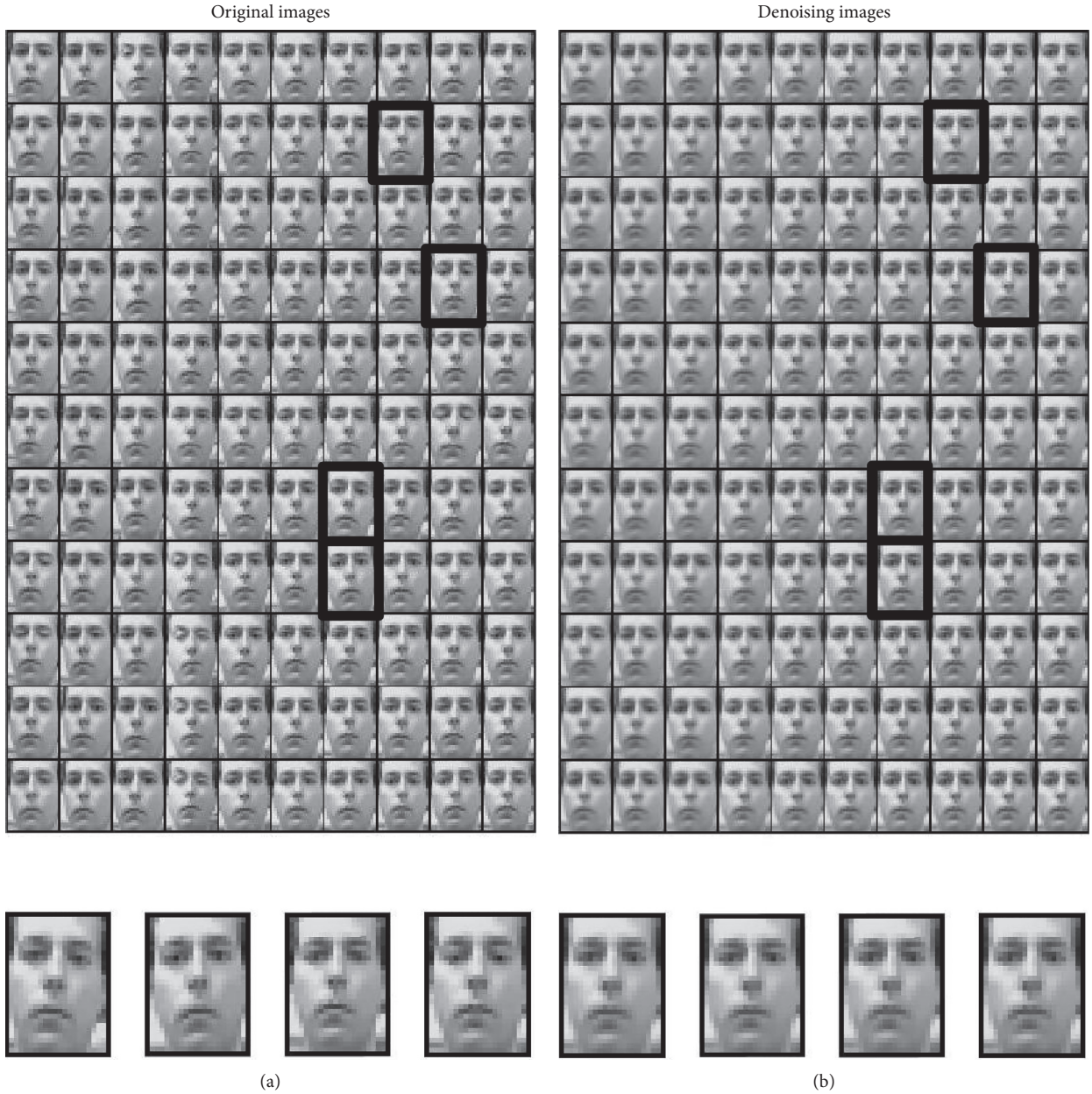

Figure 7: Denoising for the LLE face dataset. (a) Noise images. (b) Denoising images. Four corresponding typical images are listed. The posture of the head in the first two images is fixed after denoising. The head gets bigger in the third image. The eyes become open in the fourth image.

method, the smooth parameter is 0.3. The parameters in our method are set to $k = 15$, $\alpha = 3$, and $\beta = 3$.

It can be seen that three classical denoising methods have no effect on brightness noise, rotation noise, and scaling noise. These noises still exist in denoising images. The MSE even becomes larger after denoising in contrast to the noise image whereas our method has a good effect. For example, the rotation face is fixed at the fourth row and sixth column and MSE becomes smaller.

The reason is that classical image denoising methods make use of the pattern information in a single image. They could not "see" the geometric distribution information of the whole image dataset whereas our method removes noise by drawing noise data points back to the noiseless manifold-geometric distribution of the image dataset.

4.3. Denoising Dataset for Manifold Learning. In this part, we utilize our method as a preprocessing step and compare the recovered low-dimensional embeddings of noise and denoising datasets on several manifold learning algorithms. In this experiment, α, β, and k are 1, 0.8, and 13.

Figures 9(a) and 9(b) are noise "swiss roll" dataset and the ground truth of the noise dataset. Figures 9(c) and 9(d) are embeddings of the noise and denoising dataset by Isomap. Figures 9(e) and 9(f) are embeddings of the noise and

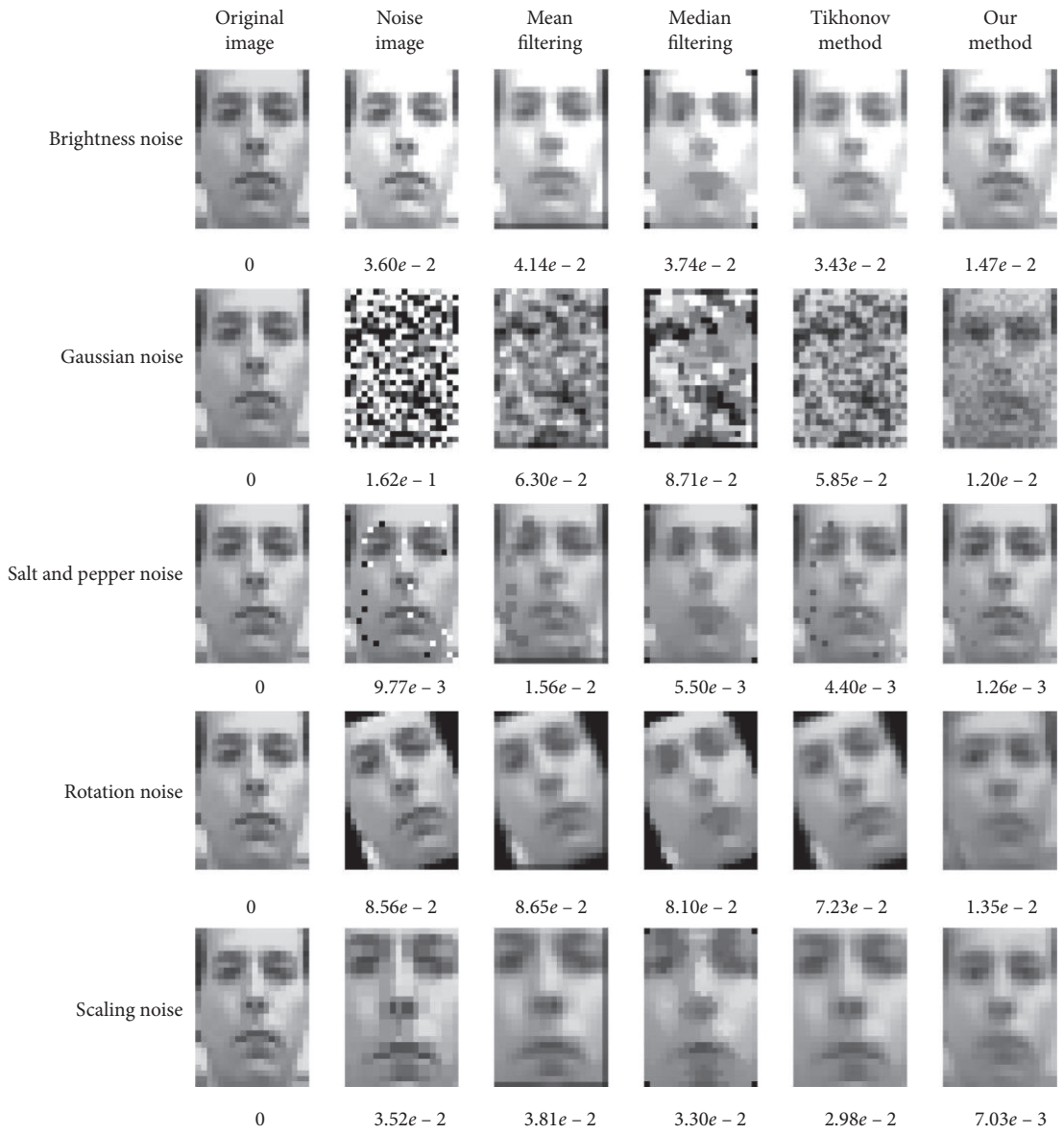

Figure 8: Denoising images' comparison. Three classical image denoising methods and our method are applied to image with five types of noise. Our method could eliminate this noise, whereas the classical image denoising methods could not deal with brightness noise, rotation noise, and scaling noise.

Table 4: MATLAB code for the noise model.

Brightness noise	NoiseImage = Image × 1.3
Gaussian noise	NoiseImage = imnoise (Image″, localvar″, size (Image)*0.5)
Salt and pepper noise	NoiseImage = imnoise (, ″salt & pepper″)
Rotation noise	NoiseImage = imrotate (Image, 20, ″bicubic″, ″*crop*″)
Scaling noise	NoiseImage = imresize (Image, 1.4, ″nearest″)

denoising dataset by LTSA. Figures 9(g) and 9(h) are embeddings of the noise and denoising dataset by HLLE. It is obvious that embeddings of the noise dataset could not reflect the geometric distribution of manifold since the neighborhoods easy to result in the "short circuit" phenomenon. By taking the denoising dataset, all the three manifold methods could get the proper embeddings. The results of Isomap result in the "hole" phenomenon because the calculated geodesic distance is always larger than it really is.

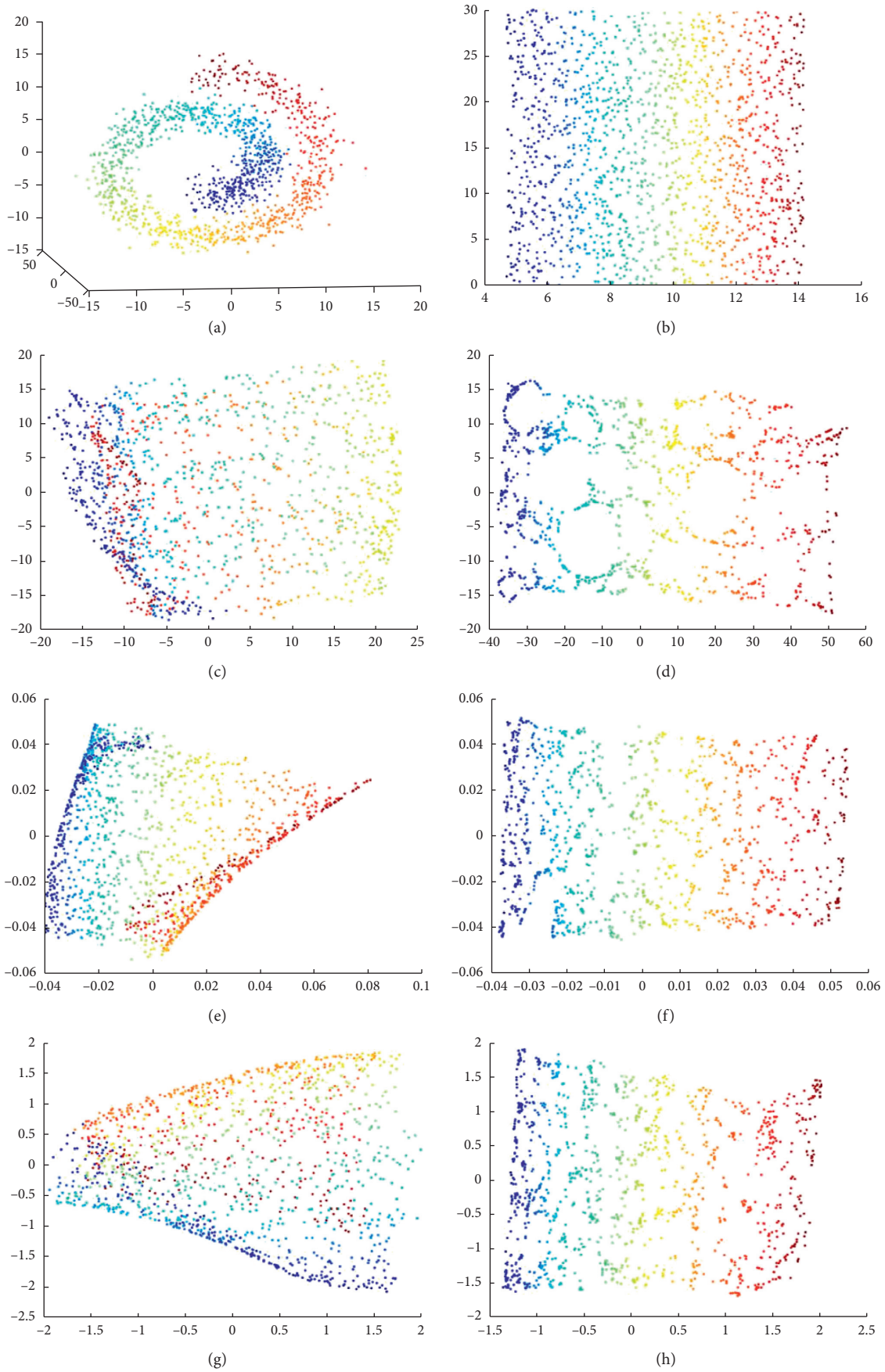

Figure 9: Embeddings of the noise dataset. (a) Noiseless dataset. (b) Ground truth. (c) and (d) Embeddings of the noise and denoising dataset by Isomap. (e) and (f) Embeddings of the noise and denoising dataset by LTSA. (g) and (h) Embeddings of the noise and denoising dataset by HLLE.

To conduct a quantitative comparison, we assess the quality of the embeddings by three indexes: embedding error, trustworthiness error, and continuity error [8]. The embedding error E measures the squared distance from the recovered low-dimensional embeddings to the ground truth coordinates which could be defined as

$$E = \sqrt{\sum_{n=1}^{N} y_n - y_n^{*2}}, \tag{23}$$

where N is the number of data points and y_n and y_n^* represent the embedding coordinates and ground true coordinates, respectively. This index tends to measure global structure distortion of the manifold.

The trustworthiness error T and continuity error C measure the local geometric structure distortion. The trustworthiness error measures the proportion of points that are too close together in the low-dimensional embedding and continuity error measures the proportion of points that are pushed away:

$$T(k) = 100 \times \frac{2}{Nk(2N - 3k - 1)} \sum_{n=1}^{N} \sum_{m \in U_n^{(k)}} (r(n,m) - k), C(k) = 100 \times \frac{2}{Nk(2N - 3k - 1)} \sum_{n=1}^{N} \sum_{m \in V_n^{(k)}} (\hat{r}(n,m) - k), \tag{24}$$

where k is the point number in the neighborhood, $r(n,m)$ is the rank of the point u_m in the ordering according to the pairwise distance from point u_n in the high-dimensional space, and $\hat{r}(n,m)$ is the rank of the point y_m in the ordering according to the pairwise distance from point y_n in low-dimensional embedding. The variables $U_n^{(k)}$ and $V_n^{(k)}$ denote the neighborhood points of u_m in low-dimensional embedding and high-dimensional space, respectively.

We test our method on several dimension reduction methods. The noise swiss roll dataset contains 1300 points. Here, we set α, β, and k to 1, 0.8, and 13. The best embedding results among several trials are selected in this experiment. The embedding error, trustworthiness error, and continuity error are listed in Tables 5–7, respectively. To show the effectiveness of our method, the errors of noise dataset, denoising dataset, and noiseless dataset are listed in three rows. It could be seen that the errors become small by taking the denoising dataset in Isomap, LLE, HLLE, LTSA, and AML. However, LE and LPP have a poor performance by taking denoising dataset.

4.4. Classification Experiment. In this part, we utilize our method as a preprocessing step and compare the accuracy rate of the original dataset and denoising dataset in the classification task. MNIST handwritten number dataset is selected which contains 60000 images with ten classes from numbers 0 to 9. Each class has about 6000 images and the size of each image is $28 * 28$ pixels. To get the denoising dataset, we utilize our denoising method for these ten classes, respectively.

In this experiment, we specify different numbers of images in each class as training data and utilize the remaining images as test data both in the original dataset and denoising dataset. A simple one-hidden-layer neural network is adopted as a classifier. The input layer has 784 units corresponding to the pixels in an image. The output layer has 10 units corresponding to ten categories from number zero to nine. We set 25 units in the hidden layer including a bias unit. The parameters of the network are trained by the BP method.

For each classification task, we repeat 10 times and list the mean accuracy rate in Figure 10. The labels "original dataset" and "denoising dataset" are raw MNIST dataset and denoising dataset with our method. The x-coordinate is the number of training images in each class and the y-coordinate is the accuracy rate. The blue and red lines are the accuracy rate of the original dataset and denoising dataset, respectively. It is obvious that the accuracy rate goes down as the number of training images decreases in each class. The performance of the denoising dataset is much better than the original dataset, especially when the training number is less than 50 in each class. The accuracy is above 96% even when there are only 10 training images in each class for the denoising dataset.

The reason is that the individual characters are removed in the denoising dataset, which is shown in Figures 5–7 in Section 3.2.1. The denoising datasets that distribute on a "clean" manifold expanded by key factors of the dataset could make machine learning algorithm easy to learn the geometric distribution knowledge of the dataset. It also illustrates that there is some kind of essential features to the classifier that is captured by our method.

Table 5: Embedding error.

	Isomap	LLE	HLLE	LTSA	LE	LPP	AML
Noise dataset	227.3	217.8	80.21	313.06	189.5	153.2	189.0
Denoising dataset	32.76	60.13	31.79	31.71	135.4	145.8	25.34
Noiseless dataset	28.79	54.63	13.42	25.80	87.29	147.9	24.70

Table 6: Trustworthiness error.

	Isomap	LLE	HLLE	LTSA	LE	LPP	AML
Noise dataset	12.78	11.46	4.43	13.91	27.48	8.39	12.84
Denoising dataset	2.99	3.29	0.94	1.23	4.12	8.34	1.09
Noiseless dataset	1.62	2.09	0.29	0.92	4.08	6.74	0.88

Table 7: Continuity error.

	Isomap	LLE	HLLE	LTSA	LE	LPP	AML
Noise dataset	5.96	3.34	1.14	4.87	5.89	2.51	4.44
Denoising dataset	1.83	2.11	0.52	0.67	2.29	2.30	0.60
Noiseless dataset	1.57	1.88	0.24	0.49	2.34	2.43	0.44

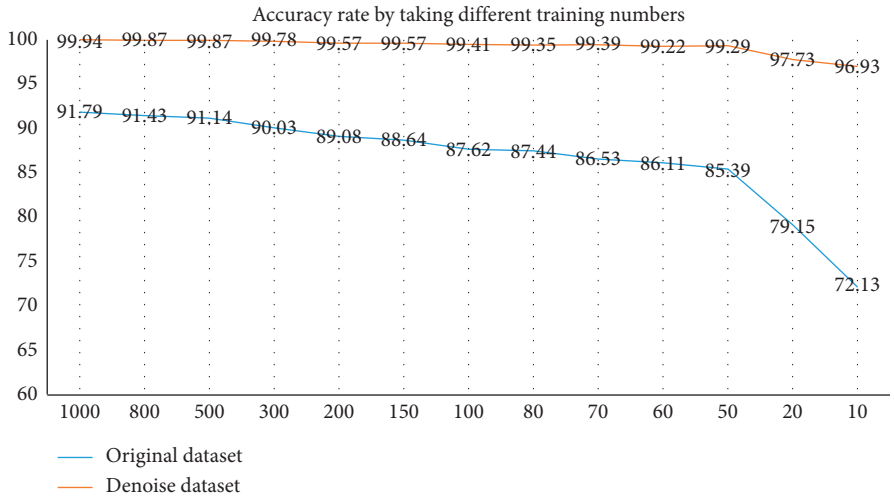

Figure 10: Accuracy rate by taking different numbers of training images.

5. Conclusion and Future Work

We propose a denoising method for the dataset rather than a single data point. This method is inspired by the manifold assumption. A local structure term is added in the Tikhonov model to make the noise points diffuse on the tangent space of the manifold. Our method could prominent the major factors hidden in the dataset and remove characteristics of the individual data point. Experiments show that our method could eliminate noise effectively on both synthetic scatter point cloud dataset and real image dataset. And as a preprocessing step, our method could improve the robustness of manifold learning and increase the accuracy rate of the classification problem. However, the parameters are sensitive in this model because the optimal solution is calculated by iteration. The geometric distribution of the dataset is distorted when the smooth term parameter is large. On the contrary, the noise intensity is still large after denoising. Our future work will focus on this problem.

Data Availability

Some or all data, models, or codes generated or used during the study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 51705304 and 61671285) and the Natural Science Foundation of Shanghai (Grant no. 19ZR1420800).

References

[1] H. S. Seung and D. D. Lee, "COGNITION: the manifold ways of perception," *Science*, vol. 290, no. 5500, pp. 2268-2269, 2000.

[2] Columbia University Image Library (COIL-20). https://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[3] J. B. Tenenbaum, V. D. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[4] S. T. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing Systems*, vol. 14, no. 6, pp. 585–591, 2002.

[6] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[7] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 253–365, 2012.

[8] R. Wang, S. Shan, X. Chen, J. Chen, and W. Gao, "Maximal linear embedding for dimensionality reduction," *Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1776–1792, 2011.

[9] Bo Zhu, J. Z. Liu, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain transform manifold learning[J]," *Nature*, vol. 7697, no. 555, pp. 487–492, 2018.

[10] S. Rahimi, A. Ali, and M. Ezoji, "Human action recognition by Grassmann manifold learning," in *Proceedings of the 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 61–64, Tehran, Iran, November 2015.

[11] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Machine Learning*, vol. 7, pp. 2399–2434, 2006.

[12] W. Liu, X. Ma, Y. Zhou, D. Tao, and J. Cheng, "p-Laplacian regularization for scene recognition," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2927–2940, 2019.

[13] P. B. Pratik, D. Wu, and Y. She, "Why deep learning works: a manifold disentanglement perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 1997–2008, 2016.

[14] X. Gu, F. Luo, J. Sun, and S.-T. Yau, "Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampère equations," *Asian Journal of Mathematics*, vol. 20, no. 2, pp. 383–398, 2016.

[15] N. Lei, D. An, Y. Guo et al., "A geometric understanding of deep learning," *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.

[16] Z. Hao, J. Liu, S. W. Ma, Xin Jin, and Xin Lian, "Noise-removal method for manifold learning," in *Proceedings of the International Conference on Life System Modeling and Simulation*, pp. 191–200, Phuket, Thailand, August 2017.

[17] Y. B. Tang, Y. Chen, N. Xu, A. Jiang, and L. Zhou, "Image denoising via sparse coding using eigenvectors of graph Laplacian," *Digital Signal Processing*, vol. 50, pp. 114–122, 2016.

[18] D. Wang, G. Song, and X. Tan, "Bayesian denoising hashing for robust image retrieval," *Pattern Recognition*, vol. 86, pp. 134–142, 2019.

[19] D. Gong, F. Sha, and G. Medioni, "Locally linear denoising on image manifolds," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 265–272, Sardinia, Italy, May 2010.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[21] W. Zhu, "Nonlocal variational methods in image and data processing," University of California, Los Angeles, CA, USA, Doctor of Philosophy in Mathematics, 2017.

[22] G. Gilboa and O. Stanley, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*vol. 7, no. 3, pp. 1005–1028, 2008.

[23] The MNIST Database of Handwritten Digits. https://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.