Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 7196043, 14 pages
http://dx.doi.org/10.1155/2016/7196043

Research Article

Nonlinear Alignment and Its Local Linear Iterative Solution

Sumin Zhang,[1,2] Zhengming Ma,[1] and Zengrong Zhan[3]

[1] *School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China*
[2] *College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China*
[3] *School of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou 511483, China*

Correspondence should be addressed to Zhengming Ma; issmzm@mail.sysu.edu.cn

Received 16 March 2016; Revised 26 July 2016; Accepted 3 August 2016

Academic Editor: Erik Cuevas

In manifold learning, the aim of alignment is to derive the global coordinate of manifold from the local coordinates of manifold's patches. At present, most of manifold learning algorithms assume that the relation between the global and local coordinates is locally linear and based on this linear relation align the local coordinates of manifold's patches into the global coordinate of manifold. There are two contributions in this paper. First, the nonlinear relation between the manifold's global and local coordinates is deduced by making use of the differentiation of local pullback functions defined on the differential manifold. Second, the method of local linear iterative alignment is used to align the manifold's local coordinates into the manifold's global coordinate. The experimental results presented in this paper show that the errors of noniterative alignment are considerably large and can be reduced to almost zero within the first two iterations. The large errors of noniterative/linear alignment verify the nonlinear nature of alignment and justify the necessity of iterative alignment.

1. Introduction

Several papers published in Science in 2000 started the research on manifold learning [1, 2]. From then on, manifold learning has made great progress and produced many representative algorithms as well as a lot of improvements around these representative algorithms, such as ISOMAP [2], Locally Linear Embedding (LLE) [1, 3], Hessian LLE (HLLE) [4], Local Tangent Space Alignment (LTSA) [5], Laplacian Eigenmaps (LE) [6], Diffusion Maps [7], and Maximum Variance Unfolding (MVU) [8]. Manifold learning as a way of machine learning has achieved good performance in many applications of machine learning.

It may be worth noting that although many machine learning algorithms claim to be manifold learning algorithms, they seem to have nothing to do with the topological manifolds defined in mathematics, not to mention the differential manifolds. The manifold learning algorithm proposed in this paper is based on the mathematical characteristics of topological/differential manifolds. This kind of algorithms can be divided into two stages: local homeomorphism and alignment. The mathematical foundation of local homeomorphism is based on the definition of topological manifolds, while the mathematical foundation of alignment is based on the characteristics of differential manifolds. In local homeomorphism, a manifold is divided into a finite number of overlapped local regions. Each local region is homeomorphic to an open set of Euclidean space. The local regions are called patches of manifold and the open sets homeomorphic to the patches are called local coordinates of patches or local coordinates of manifold directly. In alignment, the local coordinates are aligned in Euclidean space to form an area corresponding to the manifold. The area is called global coordinate of manifold. This paper only focuses on alignment; that is to say, we assume that the local coordinates have already been obtained during local homeomorphism and under this assumption we only study how to derive global coordinates from local coordinates.

The remaining sections are arranged as follows. In Section 2, some related works are reviewed briefly. In Section 3, the mathematical foundations of manifold learning are laid. In Section 4, the local nonlinear relation between the global and local coordinates is deduced mathematically. In Section 5, the local linear iterative alignment (LLIA) solution to nonlinear alignment is proposed. In Section 6 the experimental results are presented. In Section 7, some conclusions are given.

2. Related Works

There are three kinds of alignment in manifold learning: local coordinate alignment, patch alignment, and manifold alignment. The alignment in the algorithm proposed in this paper belongs to local coordinate alignment.

The local coordinate alignment is based on the mathematical definition of manifold. According to the mathematical definition of manifold, a manifold can be divided into a number of overlapped patches and each patch is homeomorphic to an open set of Euclidean space. The open sets are called local coordinates of manifold. The local coordinate alignment is to align the local coordinates in Euclidean space to form a larger open set which will be correspondent to the manifold globally. This larger open set is called global coordinate of manifold. The local coordinate alignment can be done by aligning local coordinates one by one on the light of geometrical intuition [9–11] or by turning alignment into an eigenvalue-solving problem with differential geometry [5, 12–15].

The patch alignment [16] is more algebraic than geometrical. Before alignment, the given data $X = \{x_1, \ldots, x_N\}$ have to be divided into overlapped patches $X_n = \{x_{n_1}, \ldots, x_{n_K}\} \subseteq X$, $n = 1, \ldots, N$. The patch alignment is then formulated as follows:

$$Y^* = \arg\min_{Y} \sum_{n=1}^{N} \operatorname{trace}\left(Y L_n Y^T\right), \tag{1}$$

where L_n is called local pattern of X_n, $n = 1, \ldots, N$. How to calculate L_n from X_n is dependent on different algorithms. For example, in LE algorithm [6],

$$\begin{aligned}\sum_{k=1}^{K} \omega_{n,n_k} \left\| y_n - y_{n_k} \right\|^2 &= \sum_{k=1}^{K} \omega_{n,n_k} \left\| Y s_{n_k} \right\|^2 \\ &= \sum_{k=1}^{K} \omega_{n,n_k} \operatorname{trace}\left(Y s_{n_k} s_{n_k}^T Y^T\right) \\ &= \operatorname{trace}\left(Y \left(\sum_{k=1}^{K} \omega_{n,n_k} s_{n_k} s_{n_k}^T\right) Y^T\right) \\ &= \operatorname{trace}\left(Y L_n Y^T\right),\end{aligned} \tag{2}$$

where $\omega_{n,n_k} = e^{-\|x_n - x_{n_k}\|^2/2\sigma^2}$ is the similarity between x_n and x_{n_k}, $L_n = \sum_{k=1}^{K} \omega_{n,n_k} s_{n_k} s_{n_k}^T$ and $s_{n_k} \in R^N$ such that the nth and n_kth components of s_{n_k} are 1 and −1, respectively; the other components are 0. More complicated schemes to calculate L_n based on the local similarity, local linearity, local geometry, and so on in X_n can be found in [17–19].

In supervised or semisupervised learning, besides the inherent attributes of data; there are discriminative labels assigned to data. Many recent researches in patch alignment try to incorporate the discriminative labels into the calculation of L_n. For example, in MPA algorithm [20], the neighbours of x_n are first divided into two groups: $x^w_{n_1}, \ldots, x^w_{n_{K_1}}$ have the same label with x_n, while $x^b_{n_1}, \ldots, x^b_{n_{K_2}}$ have different labels from x_n or have no label at all; then

$$\begin{aligned}\sum_{k=1}^{K_1} \omega^w_{n,n_k} \left\| y_n - y^w_{n_k} \right\|^2 - \beta \sum_{k=1}^{K_2} \omega^b_{n,n_k} \left\| y_n - y^b_{n_k} \right\|^2 &\\ = \operatorname{trace}\left(Y L^w_n Y^T\right) - \beta \operatorname{trace}\left(Y L^b_n Y^T\right) &\\ = \operatorname{trace}\left(Y L_n Y^T\right), &\end{aligned} \tag{3}$$

where $L^w_n = \sum_{k=1}^{K_1} \omega^w_{n,n_k} s^w_{n_k} (s^w_{n_k})^T$, $L^b_n = \sum_{k=1}^{K_2} \omega^b_{n,n_k} s^b_{n_k} (s^b_{n_k})^T$, $L_n = L^w_n - \beta L^b_n$. More schemes in this respect can be founded in [20, 21].

The manifold alignment is involved in a number of high-dimensional datasets which are taken from different manifolds. These datasets are all reduced to a low-dimensional Euclidean space and aligned according to certain rules. For example, let $X^f = \{x^f_1, \ldots, x^f_{N_f}\}$ and $X^g = \{x^g_1, \ldots, x^g_{N_g}\}$ be two different datasets and let $Y^f = \{y^f_1, \ldots, y^f_{N_f}\}$ and $Y^g = \{y^g_1, \ldots, y^g_{N_g}\}$ be the results of dimension reduction of X^f and X^g; then the framework of LE-like manifold alignment can be expressed as follows:

$$\begin{aligned}\arg\min_{Y^f, Y^g} \Bigg\{ &\sum_{i=1}^{N_f} \sum_{k=1}^{N_f} \omega^f_{i,j} \left\| y^f_i - y^f_j \right\|^2 + \sum_{i=1}^{N_g} \sum_{k=1}^{N_g} \omega^g_{i,j} \left\| y^g_i - y^g_j \right\|^2 \\ &+ \sum_{i=1}^{N_f} \sum_{k=1}^{N_g} \omega^{(f,g)}_{i,j} \left\| y^f_i - y^g_j \right\|^2 \Bigg\},\end{aligned} \tag{4}$$

where $\omega^f_{i,j}$ is the similarity between y^f_i and y^f_j, $\omega^g_{i,j}$ the similarity between y^g_i and y^g_j, and $\omega^{(f,g)}_{i,j}$ the similarity between y^f_i and y^g_j. If y^f_i and y^g_j are not matching points between X^f and X^g, $\omega^{(f,g)}_{i,j} = 0$. How to determine the matching points and how to calculate the similarity are application-dependent. More complicated manifold learning algorithms can be founded in [22–27].

3. Mathematical Foundations of Manifold Learning

Manifold learning can be divided into two stages: local homeomorphism and alignment. In this section we will elaborate the mathematical foundations of local homeomorphism and alignment.

3.1. Mathematical Foundations of Local Homeomorphism. Let M be a Hausdorff topological space; if, for all $x \in M$, there is a

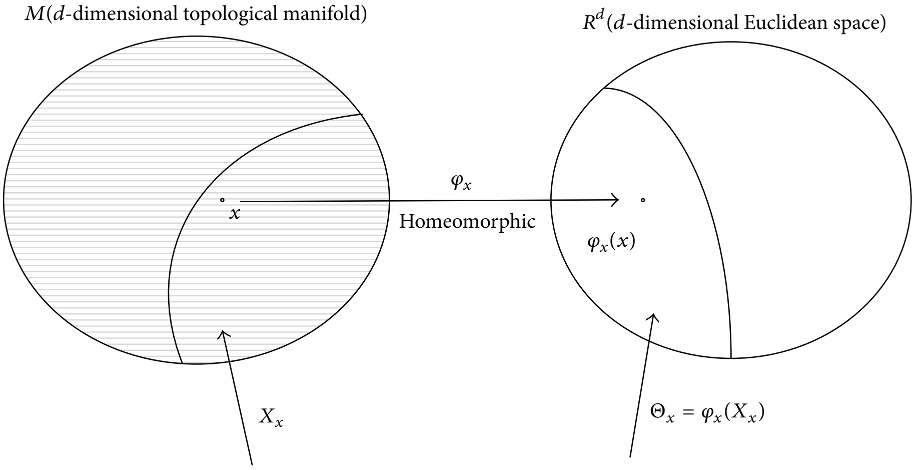

M(d-dimensional topological manifold)
R^d(d-dimensional Euclidean space)
φ_x
Homeomorphic
x
$\varphi_x(x)$
X_x
$\Theta_x = \varphi_x(X_x)$

Figure 1: Definition of topological manifolds, X_x is the neighbourhood of x, Θ_x is an open set in R^d, and φ_x is a homeomorphic mapping between X_x and Θ_x.

neighbourhood $X_x \subseteq M$ of x such that X_x is homeomorphic to an open set Θ_x of d-dimensional Euclidean space R^d, then M is called an d-dimensional topological manifold (see Figure 1). Now let $\varphi_x : X_x \to \Theta_x$ be the homeomorphic mapping between X_x and Θ_x; (X_x, φ_x) is called a chart of M and $\{(X_x, \varphi_x) \mid x \in M\}$ is called an atlas of M.

In manifold learning, the neighbourhood X_x is called a patch of M and the open set $\Theta_x = \varphi_x(X_x)$ is called the local coordinate of X_x. Furthermore, let $\widetilde{\Theta}_x = \Theta_x - \{\varphi_x(x)\}$ and $\widetilde{\varphi}_x : X_x \to \widetilde{\Theta}_x$ such that for all $x' \in X_x$, $\widetilde{\varphi}_x(x') = \varphi_x(x') - x$. It is clear that $\widetilde{\varphi}_x$ is also a homeomorphic mapping between X_x and another open set $\widetilde{\Theta}_x$. An important feature of $\widetilde{\varphi}_x$ which will facilitate the mathematical deduction of nonlinear alignment is that $\widetilde{\varphi}_x(x) = 0$. Hereafter, all the local homeomorphic mappings used in this paper are assumed to have this feature.

Furthermore, the set $\{X_x \mid x \in M\}$ is an open cover of M; that is, $M = \bigcup_{x\in M} X_x$. If M is compact, then there must be a finite subset of $\{X_x \mid x \in M\}$ such that this finite subset is also an open cover of M [28]. More specifically, if M is a d-dimensional compact manifold, then there must be a finite number of points $x_1, \ldots, x_N$ and the corresponding neighbourhoods $X_1, \ldots, X_N$ such that

(1) X_n is homeomorphic to an open set Θ_n of d-dimensional Euclidean space R^d, $n = 1, \ldots, N$.

(2) $\{X_1, \ldots, X_N\}$ is a finite open cover of M; that is, $M = \bigcup_{n=1}^{N} X_n$.

In practice, the given data $x_1, \ldots, x_N$ are always assumed to be taken from a compact manifold M and have the neighbourhoods $X_1, \ldots, X_N$ which are locally homeomorphic to the open sets $\Theta_1, \ldots, \Theta_N$ of Euclidean space. The aim of manifold learning during the stage of local homeomorphism is to find these neighbourhoods $X_1, \ldots, X_N$ and then derive the local coordinates $\Theta_1, \ldots, \Theta_N$ of these neighbourhoods. It is clear that the existence of the neighbourhoods and their local coordinates is guaranteed by the mathematical definition of topological manifold. At present, the most commonly used method of finding the neighbourhoods is the K Nearest Neighbours (KNN) method and the most commonly used method of deriving the local coordinates of neighbourhoods is the so-called tangent space method, that is, the local PCA method [5, 29].

3.2. Mathematical Foundations of Alignment. Let M be a topological manifold and $\{(X_x, \varphi_x) \mid x \in M\}$ an atlas of M; if, for any two charts (X_x, φ_x) and (X_z, φ_z) such that $X_x \cap X_z \neq \varnothing$, the mapping $\varphi_z \circ \varphi_x^{-1} : \varphi_x(X_x \cap X_z) \to \varphi_z(X_x \cap X_z)$ is C^∞ differentiable, then M is called C^∞ differential manifold (see Figure 2). In the rest part of this paper the manifolds are all assumed to be C^∞ differential manifolds.

Let $f : M \to R$ be a function defined on the differential manifold M. Generally speaking, f cannot be differentiated directly on M. In order to define the differentiability of f on M, we have to define its local pullback function first. Now let $x \in M$. According to the definition of manifold, there must be a neighbourhood $X_x \subseteq M$ of x such that X_x is homeomorphic to an open set Θ_x of Euclidean space R^d, where d is the dimension of M. Let $\varphi_x : X_x \to \Theta_x$ be the homeomorphic mapping between X_x and Θ_x, the so-called local pullback function of f is then defined as $f_x = f \circ \varphi_x^{-1} : \Theta_x \to R$ (see Figure 3).

Note that, in the theory of topological spaces, being homeomorphic means being identical. Also note that manifold is a kind of topological space. Therefore, when limited within the local regions X_x and Θ_x, f can be regarded as the same with its local pullback function $f_x = f \circ \varphi_x^{-1}$.

Furthermore, according to the definition of differential manifold, if M is C^∞ differentiable, so is the function $\varphi_z \circ \varphi_x^{-1}$. Therefore, if the function φ_z is replaced with f in $\varphi_z \circ \varphi_x^{-1}$, then the differentiability of $f \circ \varphi_x^{-1}$ is completely dependent on f. This is the reason why the differentiability of f within the local region X_x is defined by the differentiability of its local pullback function $f_x = f \circ \varphi_x^{-1}$ within the local region Θ_x. More specifically, by making use of $\varphi_x(x) = 0$, that is,

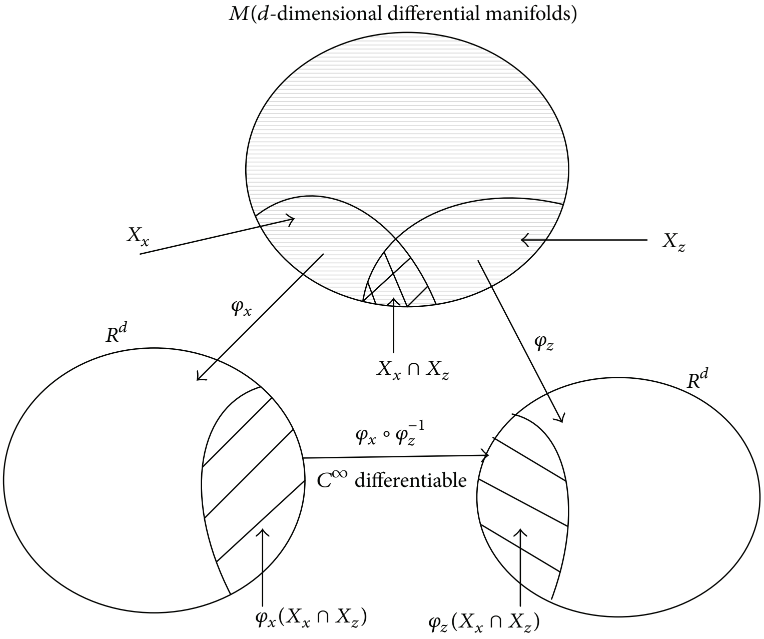

M(d-dimensional differential manifolds)
X_x
X_z
φ_x
φ_z
R^d
$X_x \cap X_z$
R^d
$\varphi_x \circ \varphi_z^{-1}$
C^∞ differentiable
$\varphi_x(X_x \cap X_z)$
$\varphi_z(X_x \cap X_z)$

Figure 2: Definition of differential manifolds: (X_x, φ_x) and (X_z, φ_z) are two charts, $X_x \cap X_z \neq \varnothing$, and M is said to be C^∞ differentiable if $\varphi_z \circ \varphi_x^{-1}$ is C^∞ differentiable.

$x = \varphi_x^{-1}(0)$, the gradient vector and Hessian matrix of f at x can be expressed as follows:

$$\begin{aligned}\nabla f(x) &= \nabla f\left(\varphi_x^{-1}(0)\right) = \nabla\left(f \circ \varphi_x^{-1}\right)(0) = \nabla f_x(0)\\ &= \left.\begin{bmatrix}\dfrac{\partial f_x(\rho)}{\partial \rho^1}\\ \vdots \\ \dfrac{\partial f_x(\rho)}{\partial \rho^d}\end{bmatrix}\right|_{\rho=0},\\ Hf(x) &= Hf\left(\varphi_x^{-1}(0)\right) = H\left(f \circ \varphi_x^{-1}\right)(0) = Hf_x(0)\\ &= \left.\begin{bmatrix}\dfrac{\partial^2 f_x(\rho)}{\partial\rho^1\partial\rho^1} & \cdots & \dfrac{\partial^2 f_x(\rho)}{\partial\rho^1\partial\rho^d}\\ \vdots & \ddots & \vdots\\ \dfrac{\partial^2 f_x(\rho)}{\partial\rho^d\partial\rho^1} & \cdots & \dfrac{\partial^2 f_x(\rho)}{\partial\rho^d\partial\rho^d}\end{bmatrix}\right|_{\rho=0}.\end{aligned} \tag{5}$$

If f is C^∞ differentiable, f can be even Taylor-expanded in the neighbourhood of x. In fact, for all $x' \in X_x$,

$$\begin{aligned}f\left(x'\right) &= f\left(\varphi_x^{-1} \circ \varphi_x\left(x'\right)\right) = \left(f \circ \varphi_x^{-1}\right)\left(\varphi_x\left(x'\right)\right)\\ &= f_x\left(\theta'\right)\\ &= f_x(0) + \left(\nabla f_x(0)\right)^T \theta' + \frac{1}{2}\theta'^T Hf_x(0)\,\theta' + \cdots\\ &= f(x) + \left(\nabla f(x)\right)^T \theta' + \frac{1}{2}\theta'^T Hf(x)\,\theta' + \cdots,\end{aligned} \tag{6}$$

where $\theta' = \varphi_x(x') \in \Theta_x$. The differentiation of local pullback functions lays the mathematical foundation of alignment.

4. Nonlinear Alignment

In manifold learning, it is always assumed that an d-dimensional manifold M can be mapped to an open set Π of the d-dimensional Euclidean space R^d. The open set Π is called the global coordinate of M. The global coordinate of a manifold is the target of manifold learning. The aim of alignment is to derive the manifold's global coordinate from its local coordinates.

Now let $\varphi : M \to \Pi$ be the mapping between the manifold M and its global coordinate Π. For all $x \in M$, $y = \varphi(x) = \left[\varphi^1(x) \ \cdots \ \varphi^d(x)\right]^T \in \Pi \subseteq R^d$ is also called the global coordinate of x, where $\varphi^i : M \to R$ is the ith component function of φ, $i = 1, \ldots, d$.

Obviously, the component functions of φ can be regarded as the functions defined on the manifold M and can be locally pulled back to the d-dimensional Euclidean space R^d (see Figure 4). According to (6), deduced in Section 3.2, the component functions can be Taylor-expanded in the neighbourhood X_x of x: for all $x' \in X_x$,

$$\begin{aligned}\varphi^i\left(x'\right) &= \varphi^i\left(\varphi_x^{-1} \circ \varphi_x\left(x'\right)\right) = \left(\varphi^i \circ \varphi_x^{-1}\right)\left(\varphi_x\left(x'\right)\right)\\ &= \varphi_x^i\left(\varphi_x\left(x'\right)\right)\\ &= \varphi^i(x) + \left(\nabla\varphi_x^i(0)\right)^T \varphi_x\left(x'\right)\\ &\quad + \frac{1}{2}\left(\varphi_x\left(x'\right)\right)^T H\varphi_x^i(0)\,\varphi_x\left(x'\right),\end{aligned} \tag{7}$$

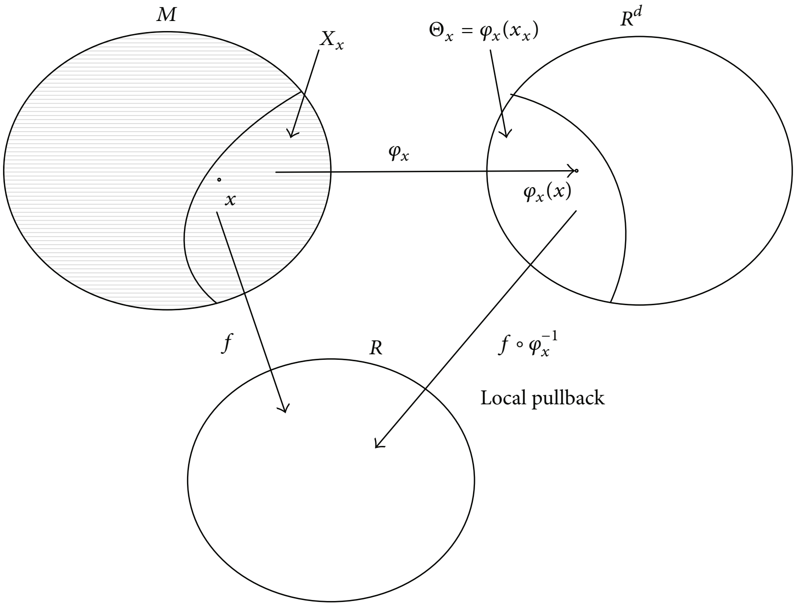


Figure 3: The Local Pullback of functions defined on manifolds; f is defined on X_x and $f \circ \varphi_x^{-1}$ is defined on Θ_x. Since X_x and Θ_x are homeomorphic, the differentiation of f on X_x can be defined by using the differentiation of its pullback function $f \circ \varphi_x^{-1}$ on Θ_x.

where $\varphi_x^i = \varphi^i \circ \varphi_x^{-1}$ is the local pullback function of φ^i from X_x into Θ_x, $i = 1, \ldots, d$. Equation (7) can be rewritten into a matrix form:

$$\begin{aligned}
y' = \varphi\left(x'\right) &= \begin{bmatrix} \varphi^1\left(x'\right) \\ \vdots \\ \varphi^d\left(x'\right) \end{bmatrix} = \begin{bmatrix} \varphi_x^1\left(\varphi_x\left(x'\right)\right) \\ \vdots \\ \varphi_x^d\left(\varphi_x\left(x'\right)\right) \end{bmatrix} \\
&= \begin{bmatrix} \varphi_x^1\left(\theta'\right) \\ \vdots \\ \varphi_x^d\left(\theta'\right) \end{bmatrix} \\
&= \begin{bmatrix} \varphi^1\left(x\right) \\ \vdots \\ \varphi^d\left(x\right) \end{bmatrix} + \begin{bmatrix} \left(\nabla\varphi_x^1\left(0\right)\right)^T \\ \vdots \\ \left(\nabla\varphi_x^d\left(0\right)\right)^T \end{bmatrix} \varphi_x\left(x'\right) \\
&\quad + \frac{1}{2} \begin{bmatrix} \left(\varphi_x\left(x'\right)\right)^T H\varphi_x^1\left(0\right)\varphi_x\left(x'\right) \\ \vdots \\ \left(\varphi_x\left(x'\right)\right)^T H\varphi_x^d\left(0\right)\varphi_x\left(x'\right) \end{bmatrix} + \cdots \\
&= \varphi\left(x\right) + A_x\varphi_x\left(x'\right) \\
&\quad + \frac{1}{2} \begin{bmatrix} \left(\varphi_x\left(x'\right)\right)^T H\varphi_x^1\left(0\right)\varphi_x\left(x'\right) \\ \vdots \\ \left(\varphi_x\left(x'\right)\right)^T H\varphi_x^d\left(0\right)\varphi_x\left(x'\right) \end{bmatrix} + \cdots \\
&= y + A_x\theta' + \frac{1}{2} \begin{bmatrix} \left(\theta'\right)^T H\varphi_x^1\left(0\right)\theta' \\ \vdots \\ \left(\theta'\right)^T H\varphi_x^d\left(0\right)\theta' \end{bmatrix} + \cdots ,
\end{aligned} \tag{8}$$

where $A_x = \begin{bmatrix} (\nabla\varphi_x^1(0))^T \\ \vdots \\ (\nabla\varphi_x^d(0))^T \end{bmatrix} \in R^{d\times d}$. In (8), $y' = \varphi(x')$ is the global coordinate of x', while $\theta' = \varphi_x(x')$ is the local coordinate of x'. Equation (8) establishes the local relationship between y' and θ'. Generally speaking, this relationship is nonlinear.

5. The Local Linear Iterative Solution to Nonlinear Alignment

In algorithm, a matrix $X = \left[x_1 \ \cdots \ x_N\right] \in R^{D\times N}$ is given, where the column vectors of X are assumed to be taken from a d-dimensional, compact, and C^∞ differential manifold $M \subseteq R^D$ which is embedded into the D-dimensional Euclidean space R^D, where $d \ll D$; the manifold learning algorithms want to find a matrix $Y = \left[y_1 \ \cdots \ y_N\right] \in R^{d\times N}$ such that $y_i = \varphi(x_i)$ is the global coordinate of x_i, $i = 1, \ldots, N$. Under such circumstances, we construct our algorithm of local linear iterative alignment.

5.1. Linearization of Nonlinear Alignment. We first locally linearize the nonlinear relation of alignment deduced in Section 4. For each data point x_n, let $X_n = \left[x_{n_1} \ \cdots \ x_{n_K}\right] = XS_n$ be the neighbourhood of x_n, where $S_n \in R^{N\times K}$ is the selection matrix in which the n_kth element of the kth column is 1; other elements are 0, $k = 1, \ldots, K$. It is noted that

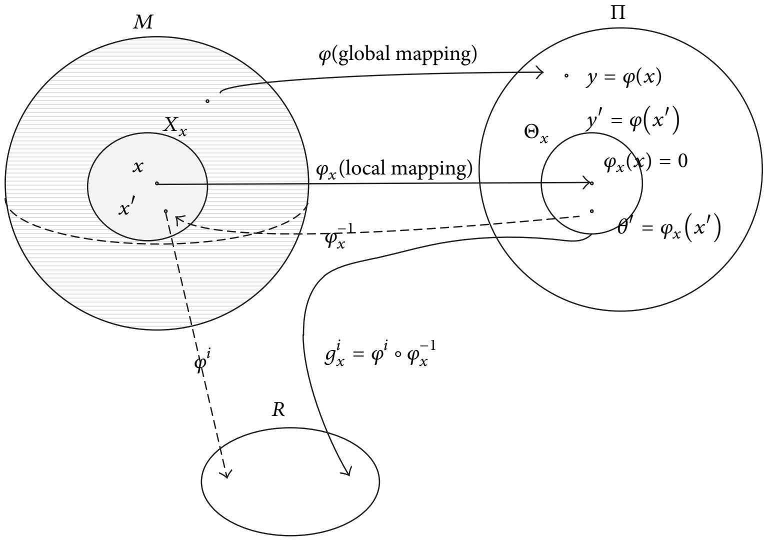

M
Π
φ(global mapping)
$y = \varphi(x)$
X_x
$y' = \varphi(x')$
Θ_x
x
φ_x(local mapping)
$\varphi_x(x) = 0$
x'
φ_x^{-1}
$\theta' = \varphi_x(x')$
φ^i
$g_x^i = \varphi^i \circ \varphi_x^{-1}$
R

Figure 4: The Local Pullback of component functions of global mapping; φ^i is the ith component function of φ and $\varphi^i \circ \varphi_x^{-1}$ is the pullback function of φ^i defined on Θ_x. Generally speaking, φ is nonlinear, so is $\varphi^i \circ \varphi_x^{-1}$. If $\varphi^i \circ \varphi_x^{-1}$ is differentiable, then $\varphi^i \circ \varphi_x^{-1}$ can be Taylor-expanded in Θ_x.

the neighbourhood X_n includes x_n itself. Let $\overline{x}_n = (1/K)\sum_{k=1}^{K} x_{n_k}$ be the centre of X_n. From now on, X_n is regarded as the neighbourhood of $\overline{x}_n$, no longer the neighbourhood of x_n.

Let $Y_n = [y_{n_1} \ \cdots \ y_{n_K}] = YS_n$ be the global coordinate of X_n, that is, $y_{n_k} = \varphi(x_{n_k})$, where $k = 1, \ldots, K$, and $\Theta_n = [\theta_{n,1} \ \cdots \ \theta_{n,K}]$ the local coordinate of X_n, that is, $\theta_{n,k} = \varphi_n(x_{n_k})$, where $k = 1, \ldots, K$ and φ_n is the local homeomorphic mapping between X_n and Θ_n. According to (8) derived in Section 4, one has

$$\begin{aligned}
y_{n_k} &= \varphi\left(x_{n_k}\right) \\
&= \varphi\left(\overline{x}_n\right) + \begin{bmatrix} \left(\nabla\varphi_n^1(0)\right)^T \\ \vdots \\ \left(\nabla\varphi_n^d(0)\right)^T \end{bmatrix} \varphi_n\left(x_{n_k}\right) \\
&\quad + \frac{1}{2}\begin{bmatrix} \left(\varphi_n\left(x_{n_k}\right)\right)^T H\varphi_n^1(0)\,\varphi_n\left(x_{n_k}\right) \\ \vdots \\ \left(\varphi_n\left(x_{n_k}\right)\right)^T H\varphi_n^d(0)\,\varphi_n\left(x_{n_k}\right) \end{bmatrix} + \cdots \\
&= \varphi\left(\overline{x}_n\right) + A_n\theta_{n,k} + \frac{1}{2}\begin{bmatrix} \theta_{n,k}^T H\varphi_n^1(0)\,\theta_{n,k} \\ \vdots \\ \theta_{n,k}^T H\varphi_n^d(0)\,\theta_{n,k} \end{bmatrix} + \cdots, \\
&\qquad k = 1, \ldots, K,
\end{aligned} \tag{9}$$

where $\varphi_n^i = \varphi^i \circ \varphi_n^{-1}$ is the local pullback function of φ^i from X_n to Θ_n, $i = 1, \ldots, d$.

Equation (9) is a nonlinear equation. From the perspective of computational mathematics, a nonlinear equation can be solved by linear iteration. Therefore (9) has to be first linearized:

$$y_{n_k} \approx \overline{y}_n + A_n\theta_{n,k}, \quad k = 1, \ldots, K, \tag{10}$$

where $\overline{y}_n = (1/K)\sum_{k=1}^{K} y_{n_k}$ is the linear part of $\varphi(\overline{x}_n)$. The error incurred from the linearization will be reduced again and again during the linear iteration.

Equation (10) can be rewritten in matrix form:

$$\begin{aligned}
\widehat{Y}_n &= [y_{n_1} \ \cdots \ y_{n_K}] - [\overline{y}_n \ \cdots \ \overline{y}_n] = Y_n C_K \\
&\approx [A_n\theta_{n,1} \ \cdots \ A_n\theta_{n,K}] = A_n[\theta_{n,1} \ \cdots \ \theta_{n,K}] \\
&= A_n\Theta_n,
\end{aligned} \tag{11}$$

where $C_K = I_K - (1/K)\Gamma_K\Gamma_K^T$ is the centralizing matrix, $\Gamma_K = [1 \ \cdots \ 1]^T \in R^K$. In geometry, (11) means that Y_n, the global coordinate of the patch X_n, can be locally approximated by translation, rotation, and scaling of Θ_n, the local coordinate of the patch X_n.

5.2. Local Linear Iterative Alignment (LLIA) Algorithm. In the proposed LLIA algorithm, the global coordinate Y is approximated iteratively:

$$\begin{aligned}
&Y^{(0)} \longrightarrow Y^{(1)} \longrightarrow \cdots \longrightarrow Y^{(p)} \longrightarrow Y^{(p+1)} \longrightarrow \cdots \\
&\quad \longrightarrow Y.
\end{aligned} \tag{12}$$

In the above iteration the initial iterative value $Y^{(0)}$ is locally set to be the local coordinates $Y_n^{(0)} = \Theta_n$, while the other iterative values $Y^{(p+1)}$ are derived from the last iterative values $Y^{(p)}$ based on the following local linear relation of alignment:

$$\widehat{Y}_n^{(p+1)} = A_n^{(p)}\widehat{Y}_n^{(p)}, \quad p = 0, 1, 2, \ldots, \tag{13}$$

where $\widehat{Y}_n^{(p)} = Y^{(p)} S_n C_K$, $\widehat{Y}_n^{(p+1)} = Y^{(p+1)} S_n C_K$. The deduction of $Y_n^{(p+1)}$ from $Y^{(p)}$ is as follows:

$$\begin{aligned}
&\sum_{n=1}^{N} \left\| \widehat{Y}_n^{(p+1)} - A_n^{(p)} \widehat{Y}_n^{(p)} \right\|^2 \\
&= \sum_{n=1}^{N} \left\| \widehat{Y}_n^{(p+1)} \left(I_K - \left(\widehat{Y}_n^{(p)} \right)^{+} \widehat{Y}_n^{(p)} \right) \right\|^2 \\
&= \sum_{n=1}^{N} \left\| Y^{(p+1)} S_n C_K \left(I_K - \left(\widehat{Y}_n^{(p)} \right)^{+} \widehat{Y}_n^{(p)} \right) \right\|^2 \\
&= \sum_{n=1}^{N} \left\| Y^{(p+1)} L_n^{(p)} \right\|^2 \\
&= \sum_{n=1}^{N} \operatorname{tr} \left(Y^{(p+1)} L_n^{(p)} \left(L_n^{(p)} \right)^T \left(Y^{(p+1)} \right)^T \right) \\
&= \operatorname{tr} \left(Y^{(p+1)} \sum_{n=1}^{N} L_n^{(p)} \left(L_n^{(p)} \right)^T \left(Y^{(p+1)} \right)^T \right) \\
&= \operatorname{tr} \left(Y^{(p+1)} L^{(p)} \left(Y^{(p+1)} \right)^T \right) \underset{Y^{(p+1)} (Y^{(p+1)})^T = I_d}{=} \min,
\end{aligned} \tag{14}$$

where $(\widehat{Y}_n^{(p)})^{+}$ is the solution to the following problem:

$$\left\| I_d - \widehat{Y}_n^{(p)} \left(\widehat{Y}_n^{(p)} \right)^{+} \right\|^2 \underset{(\widehat{Y}_n^{(p)})^{+}}{=} \min. \tag{15}$$

The algorithm framework of iterative alignment is as follows.

The Beginning of Iteration

$p = 0$

Calculate $\widehat{Y}_n^{(p)}$:

$$\widehat{Y}_n^{(p)} = \begin{cases} \Theta_n & p = 0 \\ Y^{(p)} S_n C_K & p \geq 1, \end{cases} \quad n = 1, \ldots, N \tag{16}$$

Calculate $(\widehat{Y}_n^{(p)})^{+}$:

$$\left\| I_d - \widehat{Y}_n^{(p)} \left(\widehat{Y}_n^{(p)} \right)^{+} \right\|^2 \underset{(\widehat{Y}_n^{(p)})^{+}}{=} \min, \quad n = 1, \ldots, N \tag{17}$$

Calculate $L_n^{(p)}$:

$$L_n^{(p)} = S_n C_K \left(I_K - \left(\widehat{Y}_n^{(p)} \right)^{+} \widehat{Y}_n^{(p)} \right), \quad n = 1, \ldots, N \tag{18}$$

Calculate $L^{(p)}$:

$$L^{(p)} = \sum_{n=1}^{N} L_n^{(p)} \left(L_n^{(p)} \right)^T \tag{19}$$

Calculate $Y^{(p+1)}$:

$$\operatorname{tr} \left(Y^{(p+1)} L^{(p)} \left(Y^{(p+1)} \right)^T \right) \underset{Y^{(p+1)} (Y^{(p+1)})^T = I_d}{=} \min \tag{20}$$

Calculate $\varepsilon^{(p+1)}$:

$$\varepsilon^{(p+1)} = \operatorname{tr} \left(Y^{(p+1)} L^{(p)} \left(Y^{(p+1)} \right)^T \right) \tag{21}$$

If $\varepsilon^{(p+1)} > \varepsilon$, then $p = p + 1$ and go back to calculate $\widehat{Y}_n^{(p)}$

The End of Iteration.

6. Experimental Results

Here the experimental results of 8 manifold learning algorithms on the toy and real-world data are presented. The 8 manifold learning algorithms are the proposed local linear iterative alignment (LLIA) algorithm, LLE, HLLE, LTSA, Diffusion Map, MVU, ISOMAP, and LE. The last 7 manifold learning algorithms are implemented by using MANI, a widely used platform of manifold learning which can be downloaded from internet [30].

6.1. Toy Data. The toy data used in the experiments are Trefoil, Punctured Sphere, and Toroidal Helix. These toy data are also produced by using MANI. Figures 5, 6, and 7 show the experimental results of 8 manifold learning algorithms on three toy datasets. At present there are no widely accepted quantitative evaluation criteria in manifold learning [31]. The evaluations of manifold learning algorithms are commonly based on visual perception.

At the top of Figure 5, an image of Trefoil and some coloured data taken from Trefoil are presented. Since Trefoil is constructed by twisting a planar coil into a cubic body, a good manifold learning algorithm should be able to recover the planar coil from the given coloured data and keep the order of colours unchanged. It can be seen from Figure 5 that the proposed LLIA algorithm does a good job in this respect.

The so-called Punctured Sphere is a spherical surface whose top has been cut off. The colour of the given data shown in Figure 6 indicates the position relationship between the given data. The aim of manifold learning is to produce an overhead view of Punctured Sphere on a plane. It can be seen from Figure 6 that the proposed algorithm achieves much better results than other manifold learning algorithms.

Toroidal Helix is something like that shown at the top of Figure 7. Although Toroidal Helix is much more complicated than Trefoil, it is also constructed by twisting a planar coil into a cubic body. The aim of manifold learning is also to recover the planar coil from the given coloured data. In doing so, the order of data colour must be kept unchanged because the order of data colour indicates the position relationship of data which should not be damaged during dimension reduction. It can be seen from Figure 7 that the performance of the proposed LLIA algorithm is one of the best among 8 manifold learning algorithms.

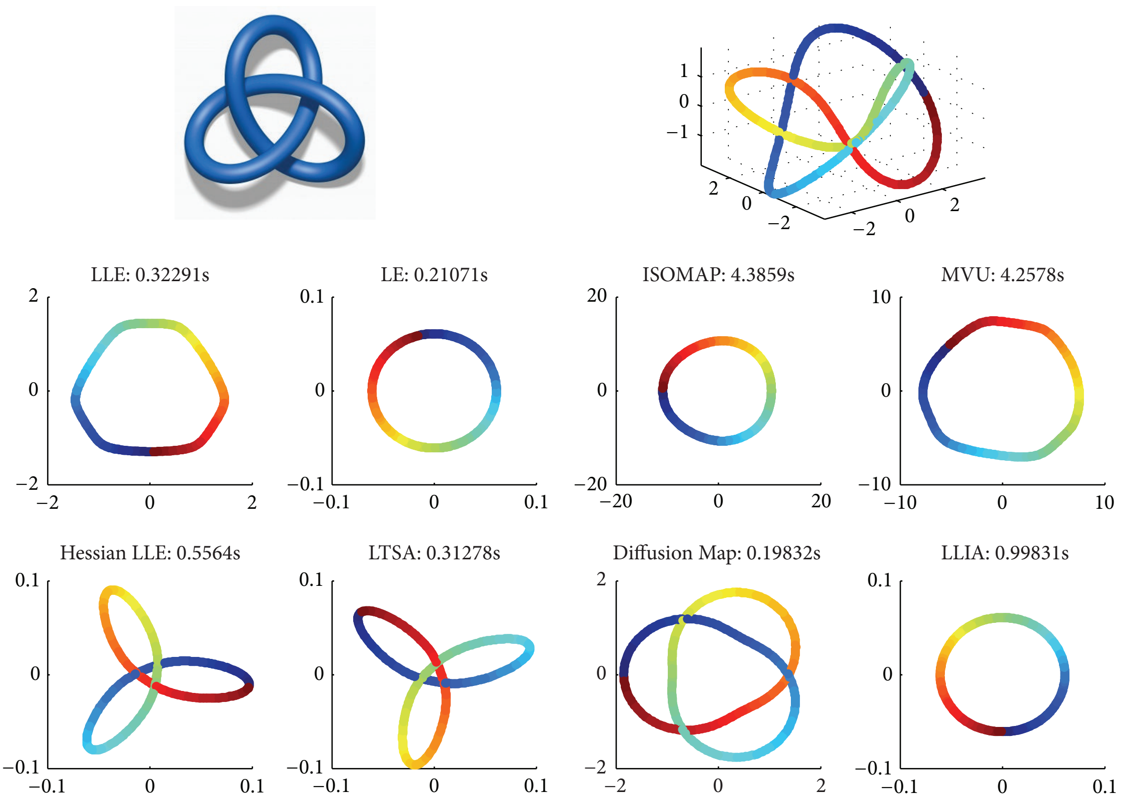


Figure 5: The experimental results on Trefoil.

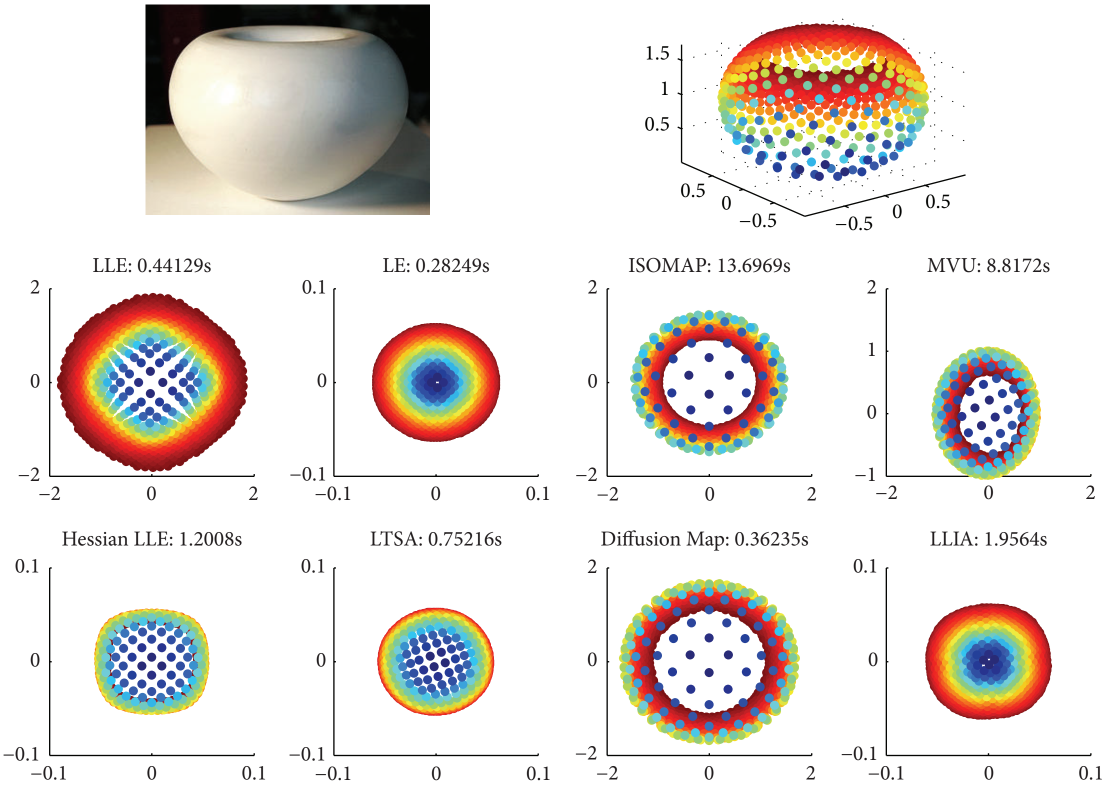


Figure 6: The experimental results on Punctured Sphere.

Table 1: Average alignment errors of LLIA algorithm on Trefoil, Punctured Sphere, and Toroidal Helix during iterations.

Iterative times	Average alignment errors (repeated 10 times)		
	Trefoil	Punctured Sphere	Toroidal Helix
1	2.430325527	6.187600	1.426720000
2	0.208952049	1.482000	0.343360000
3	$5.53147E-05$	0.000190	$2.46957E-06$
4	$5.56803E-09$	$4.85E-08$	$2.05993E-11$
5	$1.37933E-12$	$1.36E-11$	$3.60300E-14$
6	$2.2728E-14$	$3.77E-14$	$3.50165E-14$
7	$2.22939E-14$	$3.59E-14$	$3.40432E-14$
8	$2.28014E-14$	$3.56E-14$	$3.50757E-14$
9	$2.3272E-14$	$3.53E-14$	$3.47241E-14$
10	$2.22417E-14$	$3.51E-14$	$3.47867E-14$

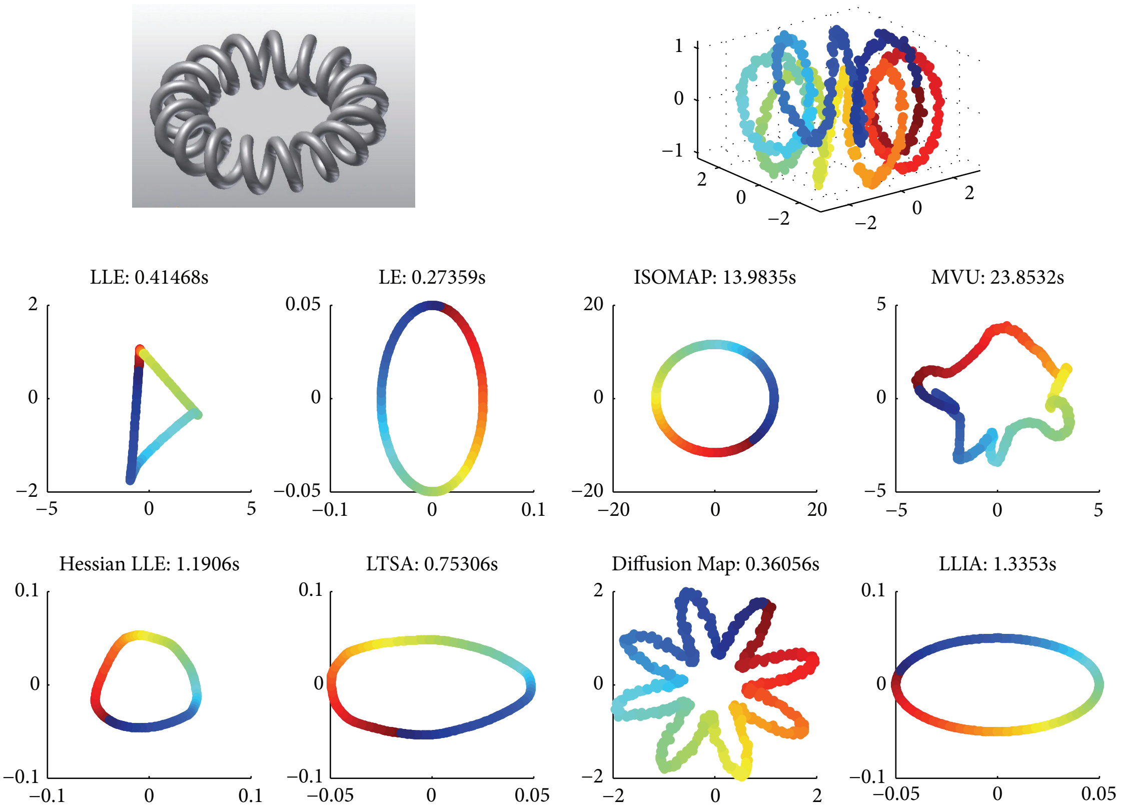


Figure 7: The experimental results on Toroidal Helix.

Table 1 shows the average alignment errors of the proposed LLIA algorithm during iterations. It can be seen from Table 1 that the initial alignment errors are considerably large and will be reduced sharply to almost zero during the first 2 iterations. The fact that the initial alignment errors are considerably large indirectly verifies the nonlinearity of alignment and justifies the necessity of iterative alignment. Furthermore the fact that the errors of iterative alignment will be reduced sharply to almost zero during the first 2 iterations means that the time complexity of the proposed LLIA algorithm is quite low.

6.2. Real-World Data. Figure 8 shows the experimental results of 8 manifold learning algorithms on the dataset of Frey Face. Frey Face is often used to test the effects of manifold learning algorithms in many academic literatures [1, 3, 32, 33]. The photos in Frey Face are all taken from the same person with different head poses and facial expressions. Therefore, although the dimension of photos is 560, the intrinsic dimension of photos is two: head pose and facial expression. In Figure 8, the photos are dimensionally reduced from 560 to 2 by using 8 different manifold learning algorithms, including the proposed LLIA algorithm. In the images of Figure 8,

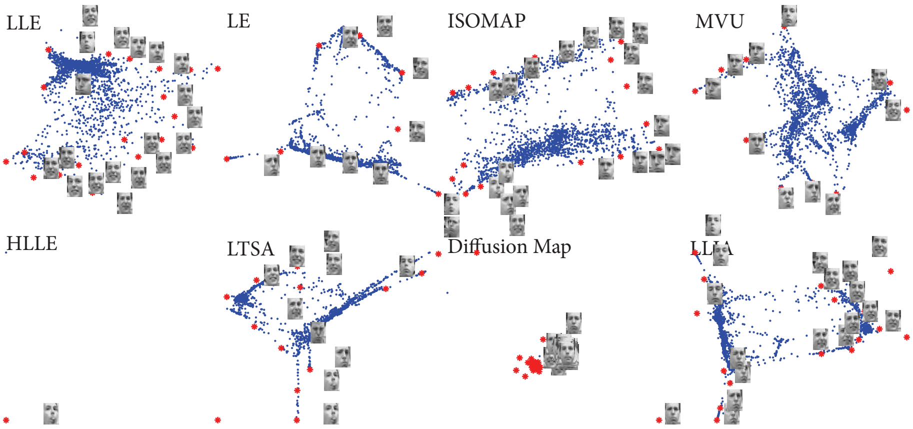


Figure 8: The experimental results on Frey Face.

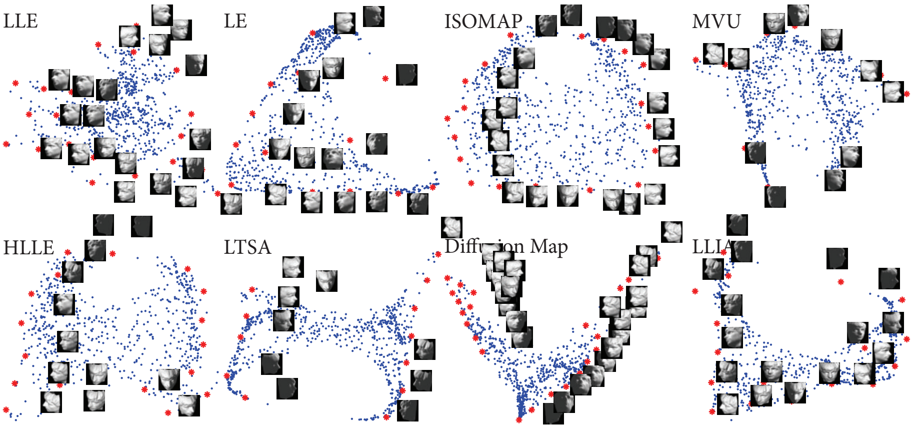


Figure 9: The experimental results on plaster.

the facial expression changes along the vertical direction, while the head pose changes along the horizontal direction. It can be seen that the proposed LLIA algorithm achieves the best visual effect among the 8 manifold learning algorithms.

Figure 9 shows the experimental results on the dataset of plaster. Plaster is also a dataset often used to test the effects of manifold learning algorithms in many academic literatures [2, 33]. The photos of plaster are all taken from the same plaster from different shooting angle and under different shooting illumination. Therefore, although the dimension of photos is $4096 = 20 \times 28$, the intrinsic dimension of photos is two: shooting angle and illumination. In Figure 9, the photos are dimensionally reduced from 4096 to 2 by using manifold learning algorithms. In the images of Figure 9, the shooting angle changes along the vertical direction, while the shooting illumination changes along the horizontal direction. Again, it can be seen that the proposed LLIA algorithm achieves the best visual effect among the 8 manifold learning algorithms.

As shown in Figures 5, 6, 7, 8, and 9, the effects of manifold learning algorithms are evaluated qualitatively, not quantitatively. At present, there are no widely accepted quantitative evaluation criteria in manifold learning [31]. However, the evaluations of manifold learning algorithms can be quantized indirectly by applying manifold learning algorithms to practical applications. In Table 2, the manifold learning algorithms are applied to the clustering application. USPS is an image dataset of handwritten numerals and has been used in many academic literatures of manifold learning [3, 34]. In Table 2, the images are first dimensionally reduced by using manifold learning algorithms and then clustered by using the K-means algorithm. The clustering precisions shown in Table 2 are then used to evaluate the effects of manifold learning algorithms indirectly. It can be seen from Figure 11 that the proposed LLIA algorithm is superior to LLE, HLLE, LTSA, and Diffusion Map but inferior to ISOMAP and LE. Note that both ISOMAP and LE belong to the so-called global preserving algorithms, while the proposed LLIA, LLE, HLLE, LTSA, and Diffusion Map belong to the so-called local preserving algorithms. Table 2 shows that, in the clustering of handwritten numerals, although the local preserving algorithms are inferior to the global preserving algorithms, the proposed LLIA achieves the best result among the local preserving algorithms.

6.3. Robustness Experiments. Figures 10 and 11 show the robustness experiments against the change of neighbourhood

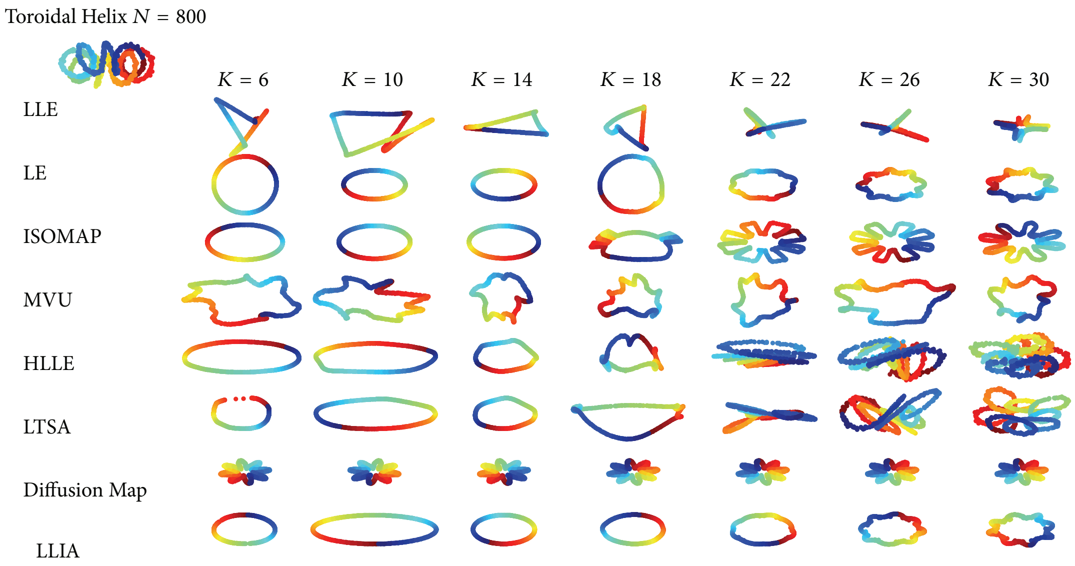


Figure 10: The robustness experiments against the change of neighbourhood sizes on Toroidal Helix.

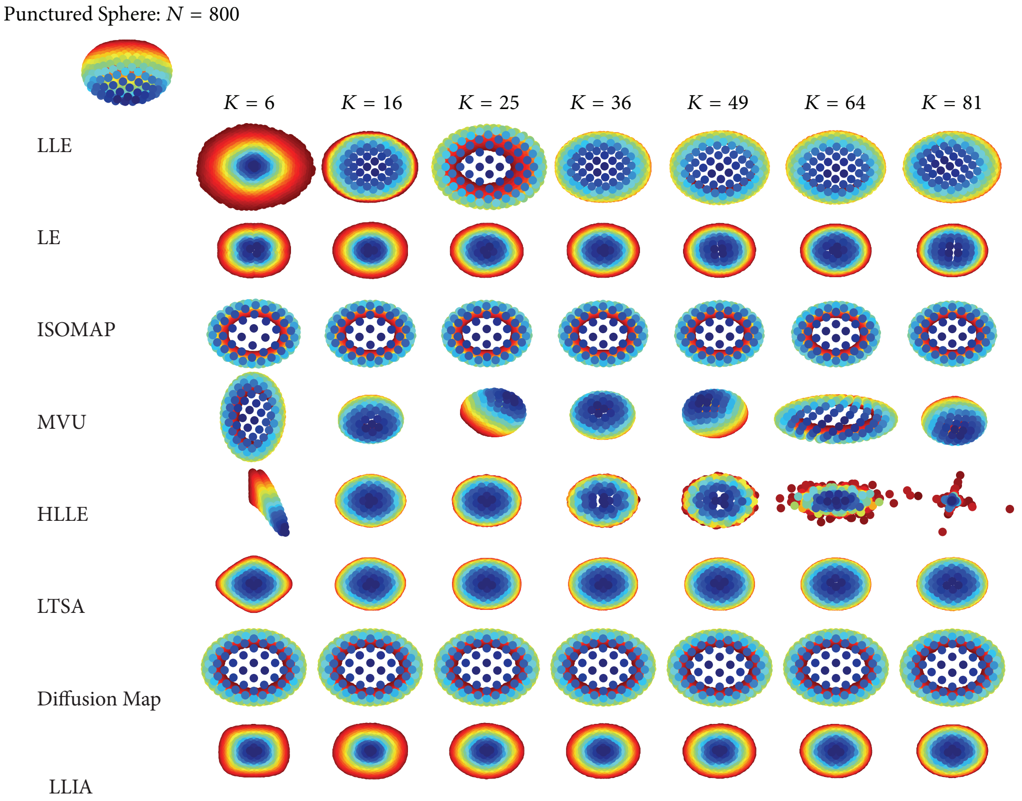


Figure 11: The robustness experiments against the change of neighbourhood sizes on Punctured Sphere.

size on Toroidal Helix and Punctured Sphere. The experimental data as well as the experimental results of manifold learning algorithms (except LLIA) are provided by using the manifold learning platform MANI.

It can be seen in Figure 10 that, on Toroidal Helix, the proposed algorithm LLIA performs best among the listed manifold learning algorithms. The experimental results of other manifold learning algorithms deform very much along with the change of K. However, as shown in Figure 11, most of the listed manifold learning algorithms (including LLIA) perform stably on Punctured Sphere. Anyway, the proposed LLIA algorithm shows some robustness against the different neighbourhood sizes as well as the different manifolds.

Figures 12 and 13 show the robustness experiments against noises on Toroidal Helix and Punctured Sphere. The noises are added by using the manifold learning platform MANI.

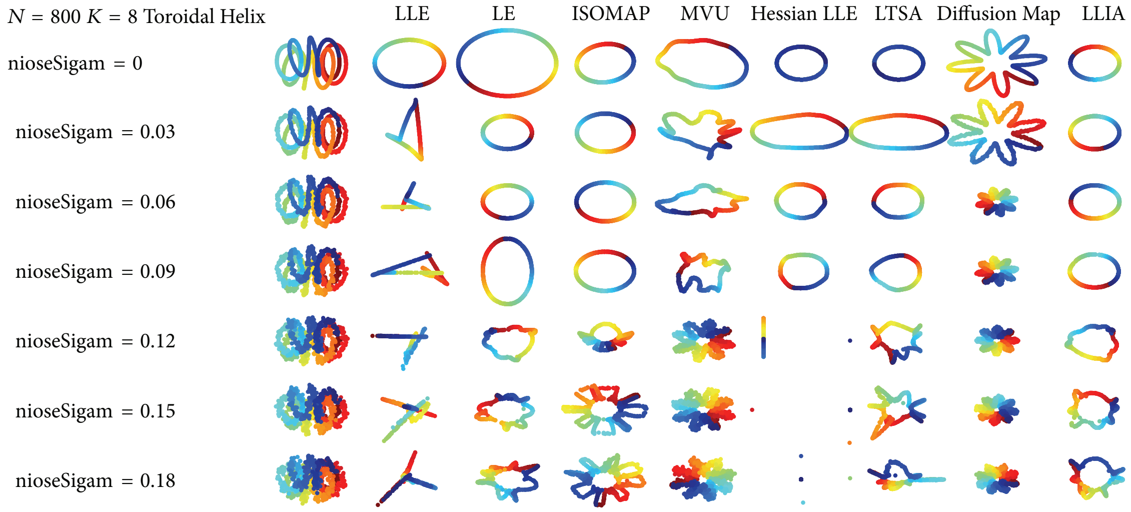


Figure 12: The robustness experiments against noises on Toroidal Helix.

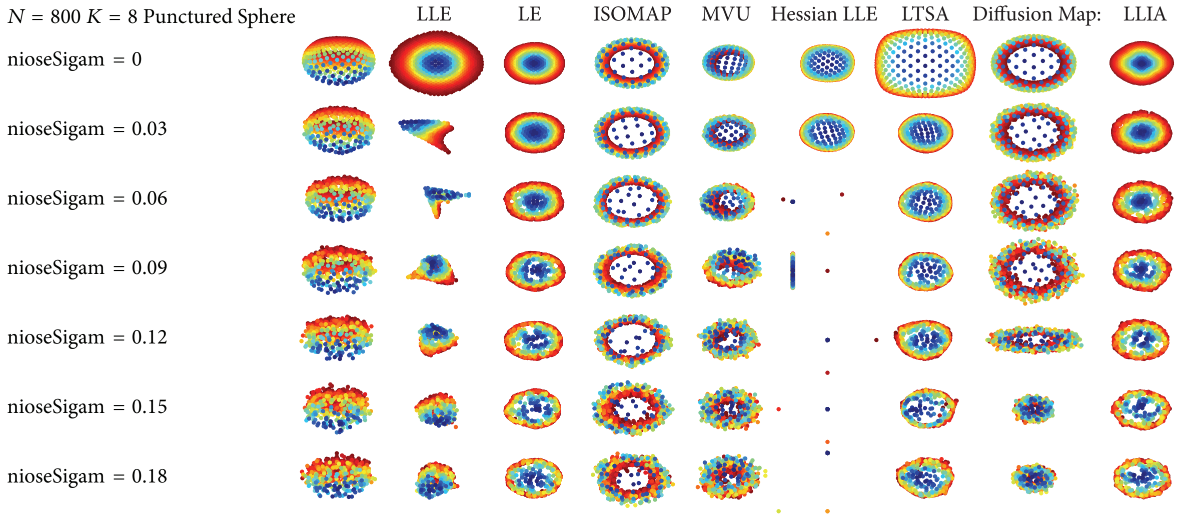


Figure 13: The robustness experiments against noises on Punctured Sphere.

Table 2: Clustering accuracy of USPS by using the 2D embedding.

	2 clusters	3 clusters	4 clusters	Average
ISOMAP	0.948000000	0.792888889	0.665888889	0.802259259
LE	0.793222222	0.705666667	0.641000000	0.713296296
LLIA	0.934333333	0.693111111	0.493333333	0.706925926
LTSA	0.879888889	0.643000000	0.505222222	0.676037037
LLE	0.821222222	0.666888889	0.508000000	0.66537037
HLLE	0.587888889	0.396777778	0.293222222	0.425962963
Diff. map	0.501333333	0.336333333	0.254111111	0.363925926

There are two Matlab sentences in MANI which can be used to add noises to the manifold data:

```
noiseSigma = handles.noise;
handles.X = handles.X + noiseSigma ∗ randn(N,3).
```

In the above Matlab sentences, the variable noiseSigma controls the noise strength. In Figures 12 and 13, the noise strength has been changed from 0 to 18%.

It can be seen in Figure 12 that, on Toroidal Helix, the performance of proposed algorithm LLIA is quite robust against noises, while other manifold learning algorithms deteriorate quickly when the noises are strengthened steadily. On the other hand, on Punctured Sphere, ISOMAP, Diffusion Map, and the proposed LLIA perform stably against the change of noises. Anyway, it can be concluded from both Figure 12 and Figure 13 that the proposed LLIA algorithm is robust against noises on different manifolds and different noise strength.

7. Conclusions

(1) At present, many machine learning algorithms, including the famous LLE algorithm [1, 35], claim to be manifold learning algorithms. However, the

manifolds mentioned in these algorithms seem to have nothing to do with the manifolds defined in mathematics. In contrast, the LLIA algorithm proposed on this paper is constructed based on the mathematical characteristics (local homeomorphism, local pullback, etc.) of differential manifolds.

(2) The manifold learning can be divided into two stages: local homeomorphism and alignment. In local homeomorphism, a Taylor expansion between a patch and its local coordinate can be deduced by using one-parameter transformation group acting on manifolds [5, 36]. In this paper, a Taylor expansion between the global and local coordinates of a patch is also deduced by using the local pullback of functions defined on differential manifold. It should be pointed out that these two Taylor expansions are completely different. The former reveals the relation between the high-dimensional data and low-dimensional date, while the latter reveals the relation between two low-dimensional data.

(3) In the proposed LLIA algorithm, the nonlinear problem is solved by using a linear iteration method. There are two problems to be addressed here: the convergence of iteration and the time complexity of algorithm. In computational mathematics, the linear iterative methods are often used to solve nonlinear problems and the convergences of iteration have been thoroughly discussed [37]. In practice, the experimental results presented in this paper show that the alignment errors would be considerably large if without iteration and will be reduced sharply during the first two iterations. This fact also implies that the time complexity of the proposed LLIA algorithm will be quite low because only few iterations are needed.

(4) At present, the algorithms of local coordinate alignment can be divided into two categories: the one-time alignment and the gradual alignment. In the one-time alignment, the manifold's global coordinate is obtained at one time by solving an eigenvalue problem [3, 5, 12–14], while in the gradual alignment, the manifold's global coordinate grows up gradually from a patch's local coordinate [9–11]. No matter which alignment is used, they are all based on the local linear model of alignment. The proposed LLIA algorithm is based on the nonlinear model of alignment and therefore belongs to neither one-time alignment nor gradual alignment.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported in part by the Guangzhou Municipal Science and Technology Projects under Grant 2014J4100095.

References

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

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

[3] J. Chen, Z. M. Ma, and Y. Liu, "Local coordinates alignment with global preservation for dimensionality reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 106–117, 2013.

[4] D. L. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10, pp. 5591–5596, 2003.

[5] 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.

[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[7] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[8] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 77–90, 2006.

[9] Z. Han, D.-Y. Meng, Z.-B. Xu, and N.-N. Gu, "Incremental alignment manifold learning," *Journal of Computer Science and Technology*, vol. 26, no. 1, pp. 153–165, 2010.

[10] Z. M. Ma and J. Chen, "The huffman-like alignment in manifold learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 4, Article ID 1451005, 2014.

[11] Y. Hou, P. Zhang, X. Xu, X. Zhang, and W. Li, "Nonlinear dimensionality reduction by locally linear inlaying," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 300–315, 2009.

[12] L. Yang, "Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 438–450, 2008.

[13] T. Zhang, X. Li, D. Tao, and J. Yang, "Local Coordinates Alignment (LCA): a novel manifold learning approach," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 4, pp. 667–690, 2008.

[14] D. Zhao, "Formulating LLE using alignment technique," *Pattern Recognition*, vol. 39, no. 11, pp. 2233–2235, 2006.

[15] Y. Zhan and J. Yin, "Robust local tangent space alignment via iterative weighted PCA," *Neurocomputing*, vol. 74, no. 11, pp. 1985–1993, 2011.

[16] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1299–1313, 2009.

[17] X. You, W. Ou, C. L. C. Chen, Q. Li, Z. Zhu, and Y. Tang, "Robust nonnegative patch alignment for dimensionality reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2760–2774, 2015.

[18] Y. Chen, Z. Jin, J. Yin, and J. Zhu, "Dimensionality reduction via locally reconstructive patch alignment," *Optical Engineering*, vol. 51, no. 7, Article ID 077208, pp. 397–407, 2012.

[19] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Transactions on Neural*

Networks & Learning Systems, vol. 25, no. 10, pp. 1779–1792, 2014.

[20] J. Xu, S. Xie, and W. Zhu, "Marginal patch alignment for dimensionality reduction," *Soft Computing*, 2015.

[21] D. Tao, L. Liang, L. Jin, and Y. Gao, "Similar handwritten Chinese character recognition using discriminative locality alignment manifold learning," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 1012–1016, IEEE, Beijing, China, September 2011.

[22] H. T. Vu, C. Carey, and S. Mahadevan, "Manifold warping: manifold alignment over time," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12)*, Toronto, Canada, July 2012.

[23] C. Wang, B. Liu, H. Vu et al., *Sparse Manifold Alignment*, 2012.

[24] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pp. 1541–1546, Barcelona, Spain, July 2011.

[25] C. Wang and S. Mahadevan, "Manifold alignment preserving global geometry," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, Beijing, China, August 2013.

[26] Z. Cui, H. Chang, S. Shan, and X. Chen, "Generalized unsupervised manifold alignment," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS '14)*, pp. 2429–2437, December 2014.

[27] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls, "Semisupervised manifold alignment of multimodal remote sensing images," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 52, no. 12, pp. 7708–7720, 2014.

[28] M. Hirsch, *Differential Topology*, Springer, Berlin, Germany, 1997.

[29] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 253–265, 2012.

[30] T. Wittman, "MANI: Manifold Learning Toolkit," http://macs.citadel.edu/wittman/research.html.

[31] J. Zhang, Q. Wang, L. He, and Z.-H. Zhou, "Quantitative analysis of nonlinear embedding," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1987–1998, 2011.

[32] S. Roweis, L. Saul, and G. Hinton, "Global coordination of local linear models," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 889–896, MIT Press, Cambridge, Mass, USA, 2001.

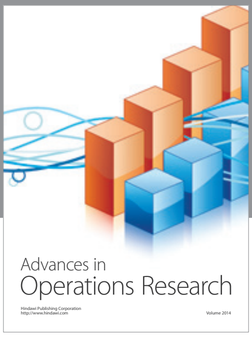
[33] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 796–809, 2008.

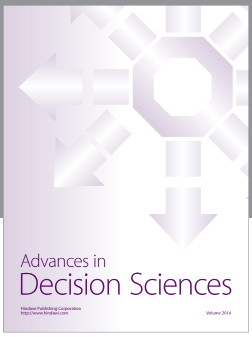
[34] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '02)*, vol. 15, pp. 857–864, 2002.

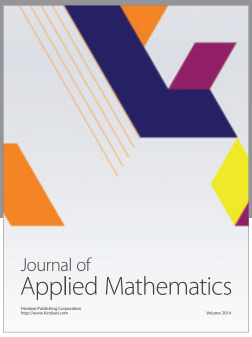
[35] J. Chen and Z. M. Ma, "Locally linear embedding: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 7, pp. 985–1008, 2011.

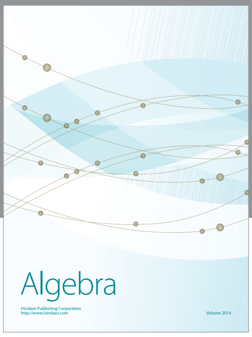
[36] W. M. Boothby, *Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, Orlando, Fla, USA, 2nd edition, 2002.

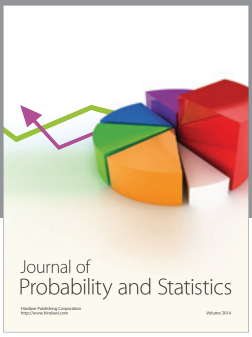
[37] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2nd edition, 2003.


Advances in
Operations Research

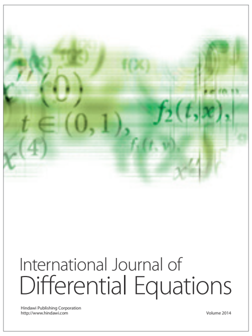

Advances in
Decision Sciences


Journal of
Applied Mathematics

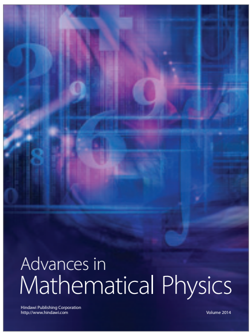

Algebra

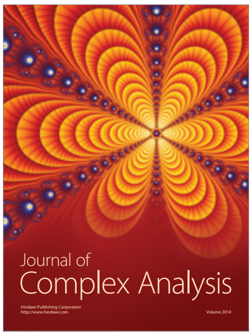

Journal of
Probability and Statistics

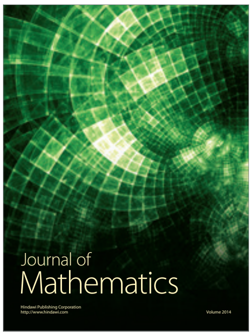
The Scientific
World Journal

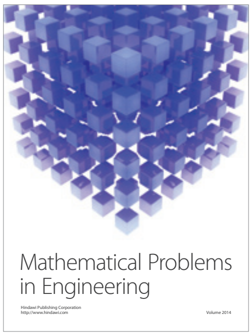

International Journal of
Differential Equations

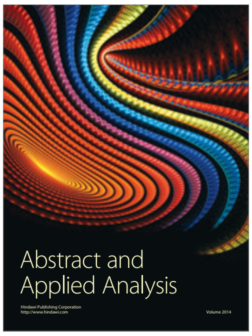
International Journal of
Combinatorics

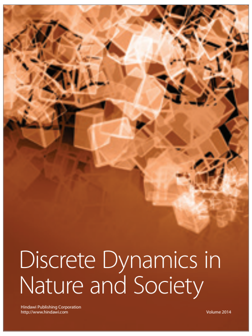

Advances in
Mathematical Physics

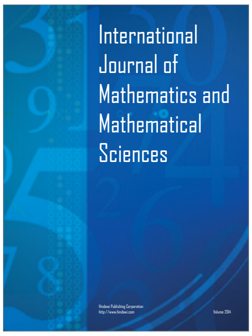

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

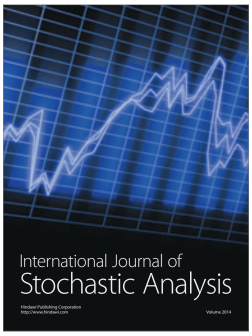

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization