Maximal Linear Embedding for Dimensionality Reduction

Ruiping Wang, *Member*, *IEEE*, Shiguang Shan, *Member*, *IEEE*, Xilin Chen, *Senior Member*, *IEEE*, Jie Chen, *Member*, *IEEE*, and Wen Gao, *Fellow*, *IEEE*

Abstract—Over the past few decades, dimensionality reduction has been widely exploited in computer vision and pattern analysis. This paper proposes a simple but effective nonlinear dimensionality reduction algorithm, named Maximal Linear Embedding (MLE). MLE learns a parametric mapping to recover a single global low-dimensional coordinate space and yields an isometric embedding for the manifold. Inspired by geometric intuition, we introduce a reasonable definition of locally linear patch, Maximal Linear Patch (MLP), which seeks to maximize the local neighborhood in which linearity holds. The input data are first decomposed into a collection of local linear models, each depicting an MLP. These local models are then aligned into a global coordinate space, which is achieved by applying MDS to some randomly selected landmarks. The proposed alignment method, called Landmarks-based Global Alignment (LGA), can efficiently produce a closed-form solution with no risk of local optima. It just involves some small-scale eigenvalue problems, while most previous aligning techniques employ time-consuming iterative optimization. Compared with traditional methods such as ISOMAP and LLE, our MLE yields an explicit modeling of the intrinsic variation modes of the observation data. Extensive experiments on both synthetic and real data indicate the effectivity and efficiency of the proposed algorithm.

Index Terms—Dimensionality reduction, manifold learning, maximal linear patch, landmarks-based global alignment.

1 Introduction

Many applications in computer vision and pattern analysis have steadily expanded their use of complex, large high-dimensional data sets. Such applications typically involve recovering compact, informative, and meaningful low-dimensional structures hidden in raw high-dimensional data for subsequent operations such as classification and visualization [24], [25], [28], [29], [37], [51], [52], [53]. An example might be a set of images of an individual's face observed under different poses and lighting conditions; the task is to identify the underlying variables given only the high-dimensional image data. Typically, the underlying structure of the observed data lies on or near a low-dimensional manifold rather than linear subspace of the (high-dimensional) input sample space. In this situation, the dimensionality reduction problem is known as "manifold learning." Generally, manifold learning approaches seek to explicitly or implicitly define a low-dimensional embedding that preserves some properties (such as geodesic distance or local relationships) of the high-dimensional observation data set.

In this paper, we propose a nonlinear dimensionality reduction algorithm, called Maximal Linear Embedding (MLE). Compared with the existing methods, MLE has several essential characteristics worth being highlighted:

1. MLE introduces a novel concept of Maximal Linear Patch (MLP), which is defined as the maximal local neighborhood in which linearity holds. The global nonlinear data structure is then represented by an integration of local linear models, each depicting an MLP.
2. MLE aligns the local models into a global low-dimensional coordinate space by a Landmarks-based Global Alignment (LGA) method, which provides an isometric embedding for the manifold. The proposed LGA method can preserve both the *local geometry* and the *global structure* of the manifold well.
3. MLE learns a nonlinear, invertible mapping function in closed form, with no risk of local optima during its global alignment procedure. Thus, the mapping can *analytically* project both training and unseen testing samples.
4. MLE is able to explicitly model the underlying modes of variability of the manifold, which has been less investigated in previous work.
5. MLE is computationally efficient. The proposed learning method is noniterative and only needs to solve an eigenproblem scaling with the number of the local models rather than the number of the training samples.

The rest of the paper is organized as follows: A brief review of dimensionality reduction methods is outlined in Section 2.

- *R. Wang is with the Broadband Network and Multimedia Lab, Department of Automation, Tsinghua University, Room 725, Central Main Building, Beijing 100084, P.R. China. E-mail: rpwang@mail.tsinghua.edu.cn.*
- *S. Shan and X. Chen are with the Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, No. 6, Kexueyuan Nanlu, Beijing 100190, P.R. China. E-mail: {sgshan, xlchen}@ict.ac.cn.*
- *J. Chen is with the Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, PL4500, Oulu FI-90014, Finland. E-mail: jiechen@ee.oulu.fi.*
- *W. Gao is with the Key Laboratory of Machine Perception (MoE), School of EECS, Peking University, Beijing 100871, P.R. China. E-mail: wgao@jdl.ac.cn.*

Manuscript received 29 Oct. 2009; revised 19 Aug. 2010; accepted 25 Nov. 2010; published online 23 Feb. 2011.
Recommended for acceptance by Y. Ma.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-10-0728.
Digital Object Identifier no. 10.1109/TPAMI.2011.39.

Section 3 describes the motivation and basic ideas of the proposed MLE. The detailed implementation of MLE along with further discussion is given in Section 4. In Section 5, extensive experiments are conducted on both synthetic and real data to evaluate the method. Finally, we give concluding remarks and a discussion of future work in Section 6.

2 Related Work

Over the past two decades, a large family of algorithms, stemming from different literatures, has been proposed to address the problem of dimensionality reduction. Among them, two representative linear techniques are principal component analysis (PCA) [20] and multidimensional scaling (MDS) [8]. In the case of so-called classical scaling, MDS is equivalent to PCA (up to a linear transformation) [37]. Recently, from the viewpoint of manifold learning, some new linear methods have been proposed, such as locality preserving projections (LPP) [18], neighborhood preserving embedding (NPE) [17], local discriminant embedding (LDE) [7], unsupervised discriminant projection (UDP) [52], and orthogonal neighborhood preserving projections (ONPP) [24]. These methods can preserve either local or global relationships and uncover the essential manifold structure within the data set.

The history of nonlinear dimensionality reduction (NLDR) traces back to Sammon's mapping [36]. Over time, other nonlinear methods have been developed, such as self-organizing maps (SOM) [23], principal curves and its extensions [16], [43], autoencoder neural networks [2], [9], and generative topographic maps (GTM) [5]. Recently, kernel methods [31], [38] provide new means to perform linear algorithms in an implicit higher-dimensional feature space. Although these methods improve the performance of linear ones, most of them are computationally expensive, and some of them have difficulties in designing cost functions or tuning many parameters, thus limiting their utility in high-dimensional data sets.

In the past few years, a new line of NLDR algorithms has been proposed based on the assumption that the data lie on or close to a manifold [39]. In general, these algorithms all formalize manifold learning as optimizing a cost function that encodes how well certain interpoint relationships are preserved [45]. For example, isometric feature mapping (ISOMAP) [42] preserves the estimated geodesic distances on the manifold when seeking the embedding. Locally linear embedding (LLE) [34] projects points to a low-dimensional space that preserves local geometric properties. Laplacian Eigenmap [3] and Hessian LLE (hLLE) [10] estimate the Laplacian and Hessian on the manifold, respectively. Semidefinite embedding (SDE) [49] estimates local angles and distances, and then "unrolls" the manifold to a flat hyperplane. Conformal eigenmaps [40] provides angle-preserving embedding by maximizing the similarity of triangles in each neighborhood. While these methods have been presented with different motivations, some researchers have tried to formalize them within a general framework, such as the kernel PCA (KPCA) interpretation [15], the graph embedding framework [51], and the Riemannian manifold learning (RML) formulation [29]. In addition, different from the traditional "batch" training mode, several incremental learning methods [26], [55] were developed recently to facilitate the applications in which data come sequentially.

Besides the above-mentioned nonparametric embedding methods, several parametric coordination methods are proposed, including global coordination [35], manifold charting [6], locally linear coordination (LLC) [41], and coordinated factor analysis (CFA) [44], [45]. These algorithms generally integrate several local feature extractors into a single global representation. They perform the nonlinear feature extraction by minimizing an objective function. After the training procedure, they are able to derive a functional mapping which can be used to project previously unseen high-dimensional observation data into their low-dimensional global coordinates.

In view of previous work, many algorithms are hindered by the so-called out-of-sample problem, i.e., they provide embeddings only for training data but not for unseen testing data. To tackle this problem, a common solution in [4] is presented for ISOMAP, LLE, and Laplacian Eigenmap. However, as a nonparametric method, in principle, it requires storage and access to all the training data, which is costly for large high-dimensional data sets, especially when generalizing the recovered manifold structure to unseen new data. Clearly, a better solution is to derive an explicit parametric mapping function between the high-dimensional sample space and the low-dimensional coordinate space.

While finding low-dimensional embedding is the core problem of manifold learning, another essential issue is to discover the underlying structure of the observation data. This can provide useful insights into the manifold geometric structure, and help to determine "interesting" regions that need extra attention [19], [21]. To this end, previous works mainly focus on the estimation of the manifold intrinsic dimensionality [13], [27], [33]. However, this is not adequate for fully exploring the manifold structure. To infer the intrinsic modes of variability of the manifold, current methods usually can only analyze the visualized embedding results in a somewhat indirect manner [34], [42], [49], based on the assumption that the coordinate axes of the embedding space correlate with the degrees of freedom underlying the original manifold data.

Moreover, compared with their linear counterparts, most existing nonlinear manifold learning approaches show inferior computational performance since they either involve a large eigenproblem scaling with the training set size [3], [6], [34], [42] or require an iterative optimization procedure such as the EM framework [35], [44].

The proposed MLE method in this paper provides a solution to the above problems, with five distinct characteristics briefly summarized in Section 1. The details of the algorithm are described in the following sections.

3 Motivation and Basic Ideas

Trusted-set methods [6], such as ISOMAP and LLE, usually define their locally linear patches on each data point by k-NN or ε-ball, generally of fixed and small size. Because this kind of definition cannot adaptively take into account the real structure of the neighborhood, it runs the risk of dividing a large linear patch into multiple smaller ones.

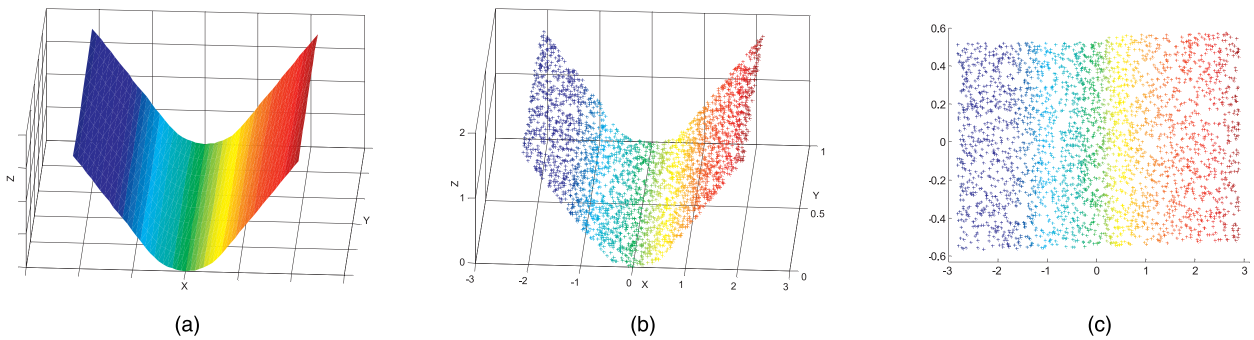


Fig. 1. The problem of nonlinear dimensionality reduction. (a) 3D "V-like shape" data, which is a patch-wise linear manifold. (b) Three thousand points are sampled from the manifold (a). (c) The proposed MLE discovers the isometric embedding in two dimensions.

Evidently, this is not economical (by *economical*, we mean to avoid excessive overlaps like in LLE). Also, it has been noted that small changes to the size of the trusted set can make the resulting embedding unstable in some cases [1]. Some efforts have been made to alleviate the effect of fixed neighborhood size [32], [48], [50]. However, the local patch definition in these methods is still essentially NN-based, without explicitly accounting for the real linear/nonlinear structure of the larger neighborhood.

In this paper, we propose to define linear patch according to the real linear/nonlinear structure in an adaptive local area. The motivation arises from some geometric intuition. See a toy example, the "V-like shape" data illustrated in Fig. 1. It is a patch-wise linear manifold, where points on the same plane actually span a "global" linear patch or subspace, and the two neighboring planes are smoothly connected. However, trusted-set methods will always ignore this "global" information. With this problem in mind, we argue that the linear patch should be defined in a more general and reasonable manner. Therefore, the concept of *Maximal Linear Patch* is introduced to capture the real linear structure. Specifically, each local patch tries to capture as much "global" information as possible and span a maximal linear subspace, whose nonlinearity degree is constrained by the deviation between the euclidean distances and geodesic distances in the patch. Fig. 2 demonstrates this idea. Intuitively, we can conjecture that each maximal linear subspace should be of the intrinsic dimensionality of the manifold.

Based on the geometric intuition of MLP, a novel hierarchical clustering algorithm is proposed to partition the sample data set into a collection of MLPs. Then, for each MLP, a local linear model can be easily computed as its low-dimensional representation by using some subspace analysis method. In this paper, PCA is exploited for this purpose considering its simplicity and analytic nature.

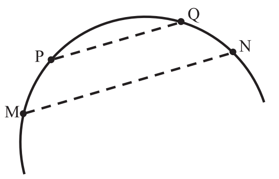


Fig. 2. Illustration of the idea of MLP. The solid semicircle represents a 1D manifold. Intuitively, the piece from P to Q is more likely to be discovered as an MLP, since its corresponding euclidean distance $\overline{PQ}$ (*dashed line*) approximates the geodesic distance $\widehat{PQ}$ (*solid arc*) preferably. In contrast, the piece from M to N is too curved to be viewed as a desirable MLP because $\overline{MN}$ (*dashed line*) deviates too much from $\widehat{MN}$ (*solid arc*).

Once the local models are constructed, we then need to align them into a global coordinate system and simultaneously seek the explicit parametric mapping. To this end, we do have some possible choices as presented in [6], [35], [41], [44], [45], etc. However, the methods in [6], [35], [44], and [45] either need the results of LLE or ISOMAP as the initialization, or are very time consuming due to the large number of local models. The method in [41] avoids such problems and provides a general solution to global alignment. However, it pursues the LLE cost function under the unit covariance constraint, which will result in the deficiency of global metrics and undesired rescaling of the manifold, as also pointed out in [29] and [30].

Therefore, we further propose a local linear model alignment method, also inspired from geometric configuration. We call the method *Landmarks-based Global Alignment*. The basic idea is as follows: We first build the global isometric coordinate system with an MDS process among a certain number of landmarks sampled sparsely from each MLP. Then, with these "locally-globally" aligned landmarks as control points, we can consistently align all the local models by estimating an explicit invertible linear transformation (translation, scale, and rotation) for each local model. By integrating these linear transformations, LGA finally results in a piecewise linear, invertible mapping function from the sample space to the global embedding space which can be naturally applied to both training and unseen testing data points.

Briefly, in sum, the main novelty of the proposed MLE is two-fold: the concept of MLP and the LGA method, which lead to several highlighted characteristics, as described in the introduction.

4 Maximal Linear Embedding

In this section, we first introduce the concept of MLP and the proposed method for MLP construction. Then, the learning procedure of MLE is presented in detail including the construction of local model, the Landmarks-based Global Alignment, i.e., the LGA method, and the analyzing method for manifold structure. Finally, comparisons of MLE with other relevant methods are discussed, followed by the complexity analysis of MLE.

4.1 Maximal Linear Patch

We can view manifold learning as an attempt to invert a generative model for a set of observation data. Given the observation data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_i \in \mathbb{R}^D$, where N is the sample number and D is the feature dimension. Assuming that these points are sampled from a manifold of intrinsic dimensionality $d < D$, we seek a nonlinear mapping onto a vector space: $F(\boldsymbol{X}) \rightarrow \boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, $\boldsymbol{y}_i \in \mathbb{R}^d$, and 1-to-1 reverse mapping $F^{-1}(\boldsymbol{Y}) \rightarrow \boldsymbol{X}$ such that both global structure and local relationships between points are preserved. As mentioned above, our method approximates the nonlinear mapping F by concatenating patch-wise local linear models, each learned from an MLP. Therefore, we first present the definition of MLP and the way to construct such MLPs from the observation data. Following that, a further discussion on a few important issues of the construction procedure is addressed.

4.1.1 MLP Construction

The principal insight for MLP lies in two criteria—1) linear criterion: for each point pair in the patch, their geodesic distance should be as close to their euclidean distance as possible, which guarantees the patch does span a near linear subspace and 2) maximal criterion: the patch size should be maximized until that any appending of additional data point would violate the linear criterion.

To construct MLPs, our earlier work [46], [47] has conducted some preliminary study on both *one-shot sequential* clustering and *hierarchical* clustering ways, mainly for the real application of object recognition with image set. In this paper, we propose to build MLPs in the more effective and flexible hierarchical manner since it allows one to create a cluster tree called dendrogram over different degrees [11], [22]. Here, for the sake of efficiency, we exploit hierarchical divisive clustering (HDC) rather than hierarchical agglomerative clustering (HAC), because in most cases the appropriate number of clusters is much smaller than the number of data samples.

Fig. 3 gives a conceptual illustration of the proposed HDC method. All samples are initiated as a singleton MLP (cluster) in the first level. Then in each new level, the MLP in the previous level with the largest nonlinearity degree will split into two smaller ones with decreased degrees. Finally, we are able to obtain multilevel MLPs associated with different nonlinearity degrees. We next formulate the algorithm in a more detailed and rigorous manner.

Formally, we aim at performing a partitioning on the data set $\boldsymbol{X}$ to obtain a collection of disjoint MLPs $\boldsymbol{X}^{(i)}$, i.e.,

$$\begin{aligned} &\boldsymbol{X} = \bigcup_{i=1}^{P} \boldsymbol{X}^{(i)}, \\ &\boldsymbol{X}^{(i)} \cap \boldsymbol{X}^{(j)} = \emptyset \quad (i \neq j, i, j = 1, 2, \ldots, P), \\ &\boldsymbol{X}^{(i)} = \{\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_{N_i}^{(i)}\} \left(\sum_{i=1}^{P} N_i = N \right), \end{aligned} \tag{1}$$

where P is the number of patches and N_i is the number of points in patch $\boldsymbol{X}^{(i)}$.

First, the pair-wise euclidean distance matrix $\boldsymbol{D}_E$ and geodesic distance matrix $\boldsymbol{D}_G$, based on k-NN graph, are computed [42]. Then a matrix holding distance ratios is obtained as: $\boldsymbol{R}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{D}_G(\boldsymbol{x}_i, \boldsymbol{x}_j)/\boldsymbol{D}_E(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Clearly, these three matrices are all of size $N \times N$. Since geodesic distance is always no smaller than euclidean distance, $\boldsymbol{R}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 1$ holds for any entry of $\boldsymbol{R}$. Besides, another matrix $\boldsymbol{H}$ of size $k \times N$ is also constructed, each column $\boldsymbol{H}(:, j)$ $(j = 1, 2, \ldots, N)$ holding the indices of k nearest neighbors of the data point $\boldsymbol{x}_j$. Note that, as a byproduct of the computation of $\boldsymbol{D}_E$ and $\boldsymbol{D}_G$, the construction of $\boldsymbol{H}$ requires no extra computation. Now we can measure the nonlinearity degree of one MLP $\boldsymbol{X}^{(i)}$ by defining a nonlinearity score function as follows:

$$S^{(i)} = \frac{1}{N_i^2} \sum_{m=1}^{N_i} \sum_{n=1}^{N_i} \boldsymbol{R}\left(\boldsymbol{x}_m^{(i)}, \boldsymbol{x}_n^{(i)}\right). \tag{2}$$

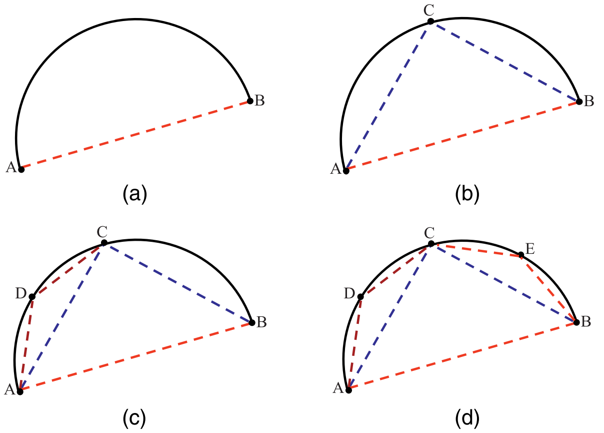


Fig. 3. Conceptual illustration of our HDC algorithm. The solid semicircle $\widehat{AB}$ represents a 1D manifold. (a)-(d) give the first four levels of MLPs output. In the first level (a), $\widehat{AB}$ is initiated as a single MLP. In the second level (b), $\widehat{AB}$ splits into two smaller ones, $\widehat{AC}$ and $\widehat{BC}$, with decreased nonlinearity degrees. In the third and fourth levels (c) and (d), $\widehat{AC}$ and $\widehat{BC}$ break into further smaller MLPs. Dashed lines in the figure represent euclidean distances between two points, and solid arcs correspond to geodesic distances.

With these definitions, the P disjoint MLPs are found using the HDC *Algorithm 1* shown in Table 1. Note that the threshold δ in step 3 controls the termination of the algorithm, and thus the number of final clusters as well as their nonlinearity degrees. Obviously, the complete clustering hierarchy can be produced whenever δ is specified to any value less than 1, since all $S^{(i)}$s are larger than 1.

4.1.2 Further Discussion

Concerning the above method for the MLP construction, several issues need to be further investigated. One is the linear criterion for MLP. Here in (2), we take the choice of the average ratio between two distances among all data pairs in a single MLP. Some alternative strategies might also be considered, such as the ratio between the respective sums of the two distances among all data pairs in the MLP, or the difference between two distances, etc. We believe that these strategies are in some sense equivalent.

Another feature is the hierarchical clustering manner in Algorithm 1. Then, how to determine an appropriate number of the final clusters (MLPs), i.e., P? Take the "V-like shape" manifold in Fig. 1 for example. By applying

TABLE 1
Algorithm 1: Hierarchical Divisive Clustering (HDC)

```
Input:  manifold data set $X = \{x_1, x_2, ..., x_N\}$
Output: a collection of disjoint MLPs $\{X^{(1)}, X^{(2)}, \dots, X^{(P)}\}$

1      Initialization: $X^{(1)} = \{x_1, x_2, ..., x_N\}$, $P = 1$; compute $S^{(1)}$ according to (2).
2      Choose $X^{(i)}$ ($i \in \{1, 2, ..., P\}$) with the largest nonlinearity score $S^{(i)}$ as the parent cluster.
3      if ($S^{(i)} > \delta$) then
3.1        According to geodesic distance matrix $D_G$, select two furthest seed points, $x_l$ and $x_r$, from $X^{(i)}$.
           Initialize the left and right child clusters: $X_l^{(i)} = \{x_l\}$, $X_r^{(i)} = \{x_r\}$. Update: $X^{(i)} \leftarrow X^{(i)} \setminus \{x_l, x_r\}$.
3.2        while ($X^{(i)} \neq \varnothing$) do
3.2.1          For current $X_l^{(i)}$, construct its neighbor points set, denote by $C_l$. According to the neighboring data
               indices matrix $H$, $C_l$ gathers the k-NN samples of all the points in $X_l^{(i)}$.
3.2.2          For current $X_r^{(i)}$, construct its neighbor points set $C_r$ in the similar way to step 3.2.1.
3.2.3          Update: $X_l^{(i)} \leftarrow X_l^{(i)} \cup (C_l \cap X^{(i)})$, $X^{(i)} \leftarrow X^{(i)} \setminus (C_l \cap X^{(i)})$; $X_r^{(i)} \leftarrow X_r^{(i)} \cup (C_r \cap X^{(i)})$, $X^{(i)} \leftarrow X^{(i)} \setminus (C_r \cap X^{(i)})$;
3.3        $X^{(i)}$ splits into: $X_l^{(i)}$ and $X_r^{(i)}$. Update: $P \leftarrow P+1$, compute $S_l^{(i)}$ and $S_r^{(i)}$ according to (2).
4      else return the current clustering results and HDC terminates.
5      Go to step 2.
```

Algorithm 1 to this data set, we can obtain the average nonlinearity score of corresponding MLPs in each clustering level, as is shown in Fig. 4a. It can be seen that, the score decreases as the levels and MLPs are increased. Fortunately, this curve provides an easy guide to select the proper number of MLPs. A simple but effective choice is the *elbow* of the curve, after the nonlinearity score falls below a reasonable value, typically being 1.1. At the elbow, the curve ceases to decrease significantly with added MLPs. In the given example, two MLPs are discovered as expected, which are demonstrated in Fig. 4b.

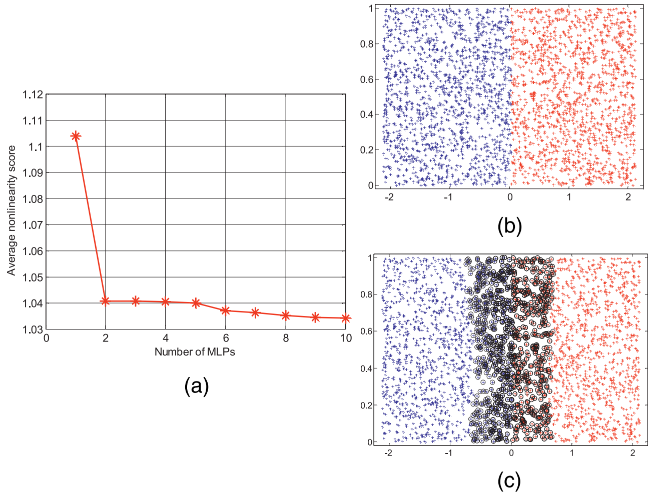


Fig. 4. Applying HDC to the "V-like shape" data. (a) The average nonlinearity score in each clustering level. (b) and (c) give the discovered MLPs in the XY view (encoded with different colors and shapes) before and after applying the soft stitching generalization, respectively. As is expected, each plane is approximately discovered as a single MLP in (b); moreover, the stitching procedure constructs some additional patches (*black open circles* in (c)) along the neighboring patch boundary.

Considering the two disjoint MLPs in Fig. 4b, one can readily raise a question that the k-NNs of those data points lying along the patch boundary are assigned to distinct MLPs. We call these data as *boundary points*. More generally, for certain types of data set (imagine a "U-like shape" manifold), it is likely to divide a large linear patch, which exactly matches to a true MLP, into two smaller clusters if only following Algorithm 1. Therefore, the algorithm cannot guarantee to finally obtain the essential MLPs, while in most real-world cases it is rather difficult or even impossible to know the true MLPs. In fact, Algorithm 1 produces a hard partitioning on the manifold. To tackle the above problem and achieve more robustness, we can further consider a soft generalization of the hard partitioning to stitch the disjoint neighboring MLPs with some additional MLPs. Specifically, each new MLP stems from a boundary point, and grows to the same nonlinearity degree as the former hard partitioning MLPs. The growing process runs in a similar way to the one-shot algorithm mentioned above. For detailed implementation, please refer to our work [46]. Fig. 4c shows the final soft partitioning results on our "V-like shape" manifold.

Clearly, the soft partitioning produces a smooth decomposition of the data set $\boldsymbol{X}$, which can lead to a more stable low-dimensional embedding space and enable the learned mapping function to be continuous to some extent. Hereinafter, we denote by M the total number of MLPs after soft partitioning.

4.2 Local Linear Models

After MLPs are obtained, we need to construct local linear model for each MLP. PCA is employed for its simplicity and efficiency. Formally, for each sample $\boldsymbol{x}_m^{(i)}$ in MLP $\boldsymbol{X}^{(i)}$, its PCA projection is computed by

$$z_m^{(i)} = \boldsymbol{W}_i^T \cdot \left(\boldsymbol{x}_m^{(i)} - \bar{\boldsymbol{x}}^{(i)}\right) \quad (m = 1, 2, \ldots, N_i, \text{ and } i = 1, 2, \ldots, M), \tag{3}$$

where the sample mean

$$\bar{\boldsymbol{x}}^{(i)} = \frac{1}{N_i} \sum_{m=1}^{N_i} \boldsymbol{x}_m^{(i)}, \tag{4}$$

and the $D \times d$ principal component matrix

$$\boldsymbol{W}_i = \left[\boldsymbol{p}_1^{(i)}, \boldsymbol{p}_2^{(i)}, \ldots, \boldsymbol{p}_d^{(i)}\right], \tag{5}$$

jointly describe the local linear model, $\mathcal{M}_i$ $(i = 1, 2, \ldots, M)$, learned from $\boldsymbol{X}^{(i)}$.

As a result, each local model $\mathcal{M}_i$ represents a local d-dimensional Cartesian coordinate system in the input sample space, centered on $\bar{\boldsymbol{x}}^{(i)}$ with axes along the column vectors of $\boldsymbol{W}_i$. Here, the dimensionality d can be determined by preserving maximal variances, and all MLPs should share a common value since they belong to the same manifold. Refer to Section 4.3.4 for more details on the estimation of d.

Local model representations of the samples in the MLP $\boldsymbol{X}^{(i)}$ then write afterward as

$$\boldsymbol{Z}^{(i)} = \left\{\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \ldots, \boldsymbol{z}_{N_i}^{(i)}\right\} \quad (i = 1, 2, \ldots, M). \tag{6}$$

4.3 Landmarks-Based Global Alignment

Now the local relationships among the samples in each MLP have been well preserved by the local models. Hence, what we need to do next is to pursue a global coordinate space that preserves the topological relationships between the local models, i.e., the global structure.

4.3.1 Landmarks Preparation

Intuitively, the global structure can be characterized by the relationships among the sample means and the principal axes of all the MLPs. So a natural choice of the final embedding space can be the isometric coordinate space learned by the MDS analysis of the sample means and some samples along the principal axes of the MLPs. Here, we name these means and sampled points *landmarks*. Evidently, the MDS must be based on geodesic distance since the relationship among the local models reflects the nonlinearity of the manifold.

In theory, to constrain each local model, we need only the mean and one sample along each principal axis, i.e., $d + 1$ landmarks. In practice, the mean is not necessarily a sample among the training set. In this case, the training sample nearest to the mean, hereinafter we call it *centroid*, is used instead. Similarly, the other landmarks need not be sampled along the principal axes. Instead, they can be randomly selected, if only their amount for each MLP is a little greater than d to ensure stability.

Formally, from each MLP $\boldsymbol{X}^{(i)}$ we randomly select a number, say n_i $(n_i \geq d + 1)$, of data points in general position as landmarks to form the following landmarks set:

$$\boldsymbol{X}_L^{(i)} = \left\{\boldsymbol{x}_{L(1)}^{(i)}, \ldots, \boldsymbol{x}_{L(n_i)}^{(i)}\right\}, \tag{7}$$

where $L(k)$ $(k = 1, 2, \ldots, n_i)$ is the original sample index in $\boldsymbol{X}^{(i)}$(refer to (1)) of the specific landmark. For convenience, the centroid sample is always set to be $\boldsymbol{x}_{L(1)}^{(i)}$.

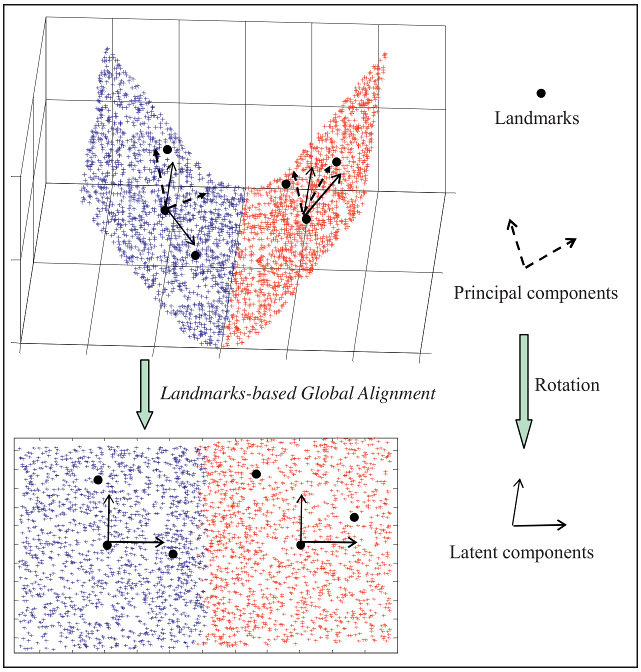


Fig. 5. Conceptual illustration of the LGA method. First, the global coordinate system is learned with an MDS process among the landmarks. Then, LGA consistently aligns all the local models by estimating a linear transformation for each local model. The transformation mainly involves a rotation from the principal components to the latent components, which are discussed in Section 4.3.4.

Denote the set of all selected landmarks by

$$\boldsymbol{X}_L = \bigcup_{i=1}^{M} \boldsymbol{X}_L^{(i)} = \left\{\boldsymbol{x}_{L(1)}^{(1)}, \ldots, \boldsymbol{x}_{L(n_1)}^{(1)}; \ldots; \boldsymbol{x}_{L(1)}^{(M)}, \ldots, \boldsymbol{x}_{L(n_M)}^{(M)}\right\}. \tag{8}$$

Correspondingly, the representations of the landmarks in their individual local model form the following set:

$$\boldsymbol{Z}_L = \bigcup_{i=1}^{M} \boldsymbol{Z}_L^{(i)} = \left\{\boldsymbol{z}_{L(1)}^{(1)}, \ldots, \boldsymbol{z}_{L(n_1)}^{(1)}; \ldots; \boldsymbol{z}_{L(1)}^{(M)}, \ldots, \boldsymbol{z}_{L(n_M)}^{(M)}\right\}. \tag{9}$$

For the ith MLP, as mentioned above, in case the sample mean $\bar{\boldsymbol{x}}^{(i)}$ is not among the training set, the centroid, say $\boldsymbol{x}_n^{(i)}$, is used instead. Then, to remain consistent, the origin of the local coordinate system must be relocated at $\boldsymbol{z}_n^{(i)}$, i.e., $\boldsymbol{z}_n^{(i)}$ should be subtracted from all the samples in $\boldsymbol{Z}^{(i)}$. Therefore, it is easy to know that, in (9), $\boldsymbol{z}_{L(1)}^{(i)}$ is a d-dimensional zero vector as follows:

$$\boldsymbol{z}_{L(1)}^{(i)} = [0, 0, \ldots, 0]^T (i = 1, 2, \ldots, M). \tag{10}$$

For notational convenience, hereinafter we still denote the centroid of each MLP by $\bar{\boldsymbol{x}}^{(i)}$.

4.3.2 Global Alignment Based on Landmarks

The landmarks can be readily exploited to pursue the global coordinate system using MDS. We then align the local models into the global space by estimating piecewise linear transformations. The procedure is intuitively illustrated in Fig. 5 and formally described next.

Given the landmarks set $\boldsymbol{X}_L$ and their corresponding interpoint geodesic distances (simply obtain from $\boldsymbol{D}_G$, refer to Section 4.1.1), classical MDS can be easily conducted to locate the landmarks uniquely in a d-dimensional euclidean space, $\mathcal{E}$. Thanks to the metric preserving property of MDS, the space $\mathcal{E}$ will then serve as a desirable destination space for isometrically embedding the whole training set $\boldsymbol{X}$. The set of the landmarks represented in $\mathcal{E}$ is written as

$$\begin{aligned}\tilde{\boldsymbol{Y}}_L &= \bigcup_{i=1}^{M} \tilde{\boldsymbol{Y}}_L^{(i)} \\ &= \{\tilde{\boldsymbol{y}}^{(1)}_{L(1)}, \dots, \tilde{\boldsymbol{y}}^{(1)}_{L(n_1)}; \dots; \tilde{\boldsymbol{y}}^{(M)}_{L(1)}, \dots, \tilde{\boldsymbol{y}}^{(M)}_{L(n_M)}\},\end{aligned} \tag{11}$$

where

$$\tilde{\boldsymbol{Y}}_L^{(i)} = \{\tilde{\boldsymbol{y}}^{(i)}_{L(1)}, \dots, \tilde{\boldsymbol{y}}^{(i)}_{L(n_i)}\}. \tag{12}$$

Toward the final embedding of the whole training data, it only remains to learn the mappings from the individual local models to the unified space $\mathcal{E}$. Let us first check the relationships between the local models and the unified space. On the one hand, the single global MDS process on all the selected landmarks implies one separate local MDS process on each MLP (up to a linear transformation). On the other hand, an MDS process on MLP is approximately equivalent to PCA (up to a linear transformation) [37] because geodesic distance is approximately equal to euclidean distance in MLP due to the nonlinearity degree constraint in (2). Consequently, we can conclude that for each MLP, the representations of its samples in PCA-based local model (i.e., $\mathcal{M}_i$) are approximately equivalent to their representations in the unified space $\mathcal{E}$, also up to a linear transformation. The three parameters of this linear transformation, i.e., rotation, translation, and scale, then need to be solved in order to map each local PCA model $\mathcal{M}_i$ to its counterpart local embedding in $\mathcal{E}$. Hereinafter, we denote the local embedding in $\mathcal{E}$ of the ith MLP by $\boldsymbol{E}_i$. Note that they have been aligned in the global space $\mathcal{E}$.

For each MLP, easy to know that $\tilde{\boldsymbol{y}}^{(i)}_{L(1)}$ should be the center of its local embedding in $\mathcal{E}$. As a result, translation can be removed by subtracting $\tilde{\boldsymbol{y}}^{(i)}_{L(1)}$ from $\boldsymbol{E}_i$. For the scale problem, it can be easily removed by scaling the coordinates in $\boldsymbol{E}_i$ to make the pair-wise distance between landmarks in $\mathcal{E}$ equal to their distance in $\boldsymbol{X}^{(i)}_L$.

Formally, we denote the landmarks in $\mathcal{E}$ after scaling and translation by

$$\hat{\boldsymbol{Y}}_L^{(i)} = \{\hat{\boldsymbol{y}}^{(i)}_{L(1)}, \dots, \hat{\boldsymbol{y}}^{(i)}_{L(n_i)}\} \quad (i = 1, 2, \dots, M), \tag{13}$$

where

$$\hat{\boldsymbol{y}}^{(i)}_{L(k)} = s_i \cdot \left(\tilde{\boldsymbol{y}}^{(i)}_{L(k)} - \tilde{\boldsymbol{y}}^{(i)}_{L(1)}\right) \quad (k = 1, 2, \dots, n_i), \tag{14}$$

with s_i being the scaling factor. Because all landmarks are embedded in the same space $\mathcal{E}$ by MDS, scaling factors for all MLPs should be the same. For the purpose of simplicity, we assume $s_i = 1$ afterward. Note that $\hat{\boldsymbol{y}}^{(i)}_{L(1)}$ becomes also a d-dimensional zero vector. Thus, the only difference between the coordinates in $\boldsymbol{Z}^{(i)}_L$ and $\hat{\boldsymbol{Y}}^{(i)}_L$ is determined by a rotation operation. As we know, this rotation can be characterized by a $d \times d$ transition matrix $\boldsymbol{T}_i$, which should be an orthogonal matrix *theoretically* and satisfy the coordinate transformation as

$$\left[\boldsymbol{z}^{(i)}_{L(1)} \cdots \boldsymbol{z}^{(i)}_{L(n_i)}\right]_{d\times n_i} = \boldsymbol{T}_i \cdot \left[\hat{\boldsymbol{y}}^{(i)}_{L(1)} \cdots \hat{\boldsymbol{y}}^{(i)}_{L(n_i)}\right]_{d\times n_i}. \tag{15}$$

Let $\boldsymbol{A}_i = [\boldsymbol{z}^{(i)}_{L(1)} \cdots \boldsymbol{z}^{(i)}_{L(n_i)}]_{d\times n_i}$ and $\boldsymbol{B}_i = [\hat{\boldsymbol{y}}^{(i)}_{L(1)} \cdots \hat{\boldsymbol{y}}^{(i)}_{L(n_i)}]_{d\times n_i}$, $\boldsymbol{T}_i$ can then be solved by

$$\boldsymbol{T}_i \doteq \boldsymbol{A}_i \boldsymbol{B}_i^{\dagger} = \boldsymbol{A}_i \boldsymbol{B}_i^T \left(\boldsymbol{B}_i \boldsymbol{B}_i^T\right)^{-1}, \tag{16}$$

where $(\cdot)^{\dagger}$ denotes pseudo-inverse. For each MLP $\boldsymbol{X}^{(i)}$, $\boldsymbol{T}_i$ therefore describes the mapping from the local PCA model $\mathcal{M}_i$ to its local embedding $\boldsymbol{E}_i$ in the global unified space $\mathcal{E}$.

Note that (16) needs to compute the inverse matrix of $\boldsymbol{B}_i\boldsymbol{B}_i^T$. Fortunately, as discussed in Section 4.3.1, the n_i ($n_i \geq d+1$) randomly selected landmarks in general position can generally ensure $rank(\boldsymbol{B}_i) \geq d$, thus guaranteeing the nonsingularity of $\boldsymbol{B}_i\boldsymbol{B}_i^T$.

The final embedding of the whole training data now can be fulfilled by applying corresponding transformation from each local linear model to the global coordinate space. The transformation only involves very simple computations as follows:

$$\begin{aligned}\boldsymbol{y}^{(i)}_m &= \boldsymbol{T}_i^{-1} \cdot \boldsymbol{z}^{(i)}_m + \tilde{\boldsymbol{y}}^{(i)}_{L(1)} \\ &= \boldsymbol{T}_i^{-1} \cdot \left(\boldsymbol{W}_i^T \cdot \left(\boldsymbol{x}^{(i)}_m - \bar{\boldsymbol{x}}^{(i)}\right)\right) + \tilde{\boldsymbol{y}}^{(i)}_{L(1)} \\ &(m = 1, 2, \dots, N_i, \text{ and } i = 1, 2, \dots, M).\end{aligned} \tag{17}$$

Grouping results from all models, according to the sample indices in the training set, we get the final d-dimensional coordinates: $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N\}$, $\boldsymbol{y}_i \in \mathbb{R}^d$. Recall that the soft partitioning in Section 4.1.2 has assigned a number of boundary points into multiple local models. In our current setting, their final coordinates are computed simply by averaging the multiresults from corresponding local models.

To summarize, so far we have learned an explicit mapping function: $F = \{F_1, F_2, \dots, F_M\}$, where F_i $(i = 1, 2, \dots, M)$ is parameterized by (17) with parameters $\{\bar{\boldsymbol{x}}^{(i)}, \boldsymbol{W}_i, \boldsymbol{T}_i, \tilde{\boldsymbol{y}}^{(i)}_{L(1)}\}$.

4.3.3 Analytic Projection of Unseen Samples

The mapping function (17) gives an explicit forward mapping from the observation space to the embedding space. Furthermore, its reverse mapping can be easily deduced in an entirely inverse manner, i.e.,

$$\begin{aligned}&\boldsymbol{x}^{(i)}_m = \bar{\boldsymbol{x}}^{(i)} + \boldsymbol{W}_i \cdot \boldsymbol{T}_i \cdot \left(\boldsymbol{y}^{(i)}_m - \tilde{\boldsymbol{y}}^{(i)}_{L(1)}\right) \\ &(m = 1, 2, \dots, N_i, \text{ and } i = 1, 2, \dots, M).\end{aligned} \tag{18}$$

Equations (17) and (18) imply another advantage of the proposed method, i.e., once the mapping function F is learned, the training set is no longer required for subsequent process, leading to significant computational and storage savings.

Easy to understand, as the mapping between the two spaces is built through a mixture of linear transformations, when applying to new test data, MLE only needs to first identify to which local model the test data belongs and then perform the corresponding transformation. Specifically, as formulated in Tables 2 and 3, two algorithms are designed to generalize the training results to unseen cases in the observation and embedding space, respectively.

TABLE 2
Algorithm 2: Visualization Algorithm

```
Input:  test data in the observation space $x_t \in \mathbb{R}^D$
Output: $d$-dimensional embedding result $y_t \in \mathbb{R}^d$
1  Assign $x_t$ to the subspace index $j$ with minimal representation error:
       $j = \arg\min_k(\|x_t - (W_k \cdot W_k^T \cdot (x_t - \bar{x}^{(k)}) + \bar{x}^{(k)})\|)$,
       $(k = 1, 2, ..., M)$                                                    (19)
2  Compute the embedding coordinates:
       $y_t = T_j^{-1} \cdot W_j^T \cdot (x_t - \bar{x}^{(j)}) + \tilde{y}_{L(1)}^{(j)}$   (20)
```

TABLE 3
Algorithm 3: Reconstruction Algorithm

```
Input:  test data in the embedding space $y_t \in \mathbb{R}^d$
Output: $D$-dimensional virtual example $x_t \in \mathbb{R}^D$
1  Assign $y_t$ to the subspace index $j$ whose center is nearest to $y_t$:
       $j = \arg\min_k(\|y_t - \tilde{y}_{L(1)}^{(k)}\|)$, $(k = 1, 2, ..., M)$   (21)
2  Compute the reconstructed virtual example:
       $x_t = \bar{x}^{(j)} + W_j \cdot T_j \cdot (y_t - \tilde{y}_{L(1)}^{(j)})$   (22)
```

4.3.4 Underlying Manifold Structure

The intrinsic dimensionality of a manifold, d, represents the underlying degrees of freedom of the observation data. Intuitively, under manifold assumption, both the local PCA models and the unified embedding space should be of d-dimension. As stated before, the dimensionality d of PCA can be roughly chosen by preserving maximal variances. On the other hand, when classical MDS is conducted to pursue the embedding space, as indicated in [42], d can also be observed from the residual variance curve. In this paper, we combine the merits of both PCA and MDS for the estimation of d in a validation-feedback fashion by deriving the following method.

First, a relatively small interval for possible d, e.g., $[d_{\min}, d_{\max}]$, can be estimated from both PCA and MDS. Then, transformation error caused by (15) is utilized as a cost function to evaluate each value in this interval. That is to say, we aim at minimizing the transformation distortions between the low-dimensional representations of PCA and that of MDS over all landmarks. Specifically, for each MLP, since the transition matrix $\boldsymbol{T}_i$ is solved from $\boldsymbol{A}_i = \boldsymbol{T}_i \cdot \boldsymbol{B}_i$, then $\boldsymbol{B}_i^* = \boldsymbol{T}_i^{-1} \cdot \boldsymbol{A}_i$ should be as close to $\boldsymbol{B}_i$ as possible. Therefore, the optimization for estimating the optimal d can be written as follows:

$$\begin{aligned} d^* &= \arg\min_d \sum_{i=1}^{M} \sum_{k=1}^{n_i} \left\| \boldsymbol{T}_i^{-1} \cdot \boldsymbol{z}_{L(k)}^{(i)} - \hat{\boldsymbol{y}}_{L(k)}^{(i)} \right\|, \\ &s.t.\ d_{\min} \le d \le d_{\max}, \end{aligned} \tag{23}$$

where $\boldsymbol{T}_i \in \mathbb{R}^{d \times d}$, $\boldsymbol{z}_{L(k)}^{(i)}, \hat{\boldsymbol{y}}_{L(k)}^{(i)} \in \mathbb{R}^{d \times 1}$. This optimization thus combines the estimations of PCA and MDS together to make the final arbitration. Because the lower dimensional coordinates of both PCA and MDS remain the same while higher ones are added, the two processes only need to be performed once. Hence, the optimization is very efficient.

With the estimated intrinsic dimensionality, one may further concern the hidden variation modes, each corresponding to one dimension or degree of freedom, to fully explore the manifold structure. Here, by hidden variation modes, we mean the directions in the high-dimensional observation space along which the manifold data exhibit *global* variability. For instance, the "V-like shape" manifold in Fig. 1 has two hidden variation modes, one along the curved direction in the XOZ plane and another along the depth direction parallel to the Y-axis. To deduce such modes, previous work usually can only act in an indirect manner [34], [42], [49], by visualizing and analyzing the distribution of training data in the embedding space.

In contrast, our method enables an explicit modeling of the hidden variation modes. Let us revisit Fig. 5. Within each MLP, the PCA basis $\boldsymbol{W}_i$ (i.e., principal components, shown as the dashed line axes) describes the directions with the largest variances confined only to that local region. To characterize the global variations across different MLPs, the PCA basis needs to be transformed to another basis $\boldsymbol{W}_i^{\mathcal{E}}$ (shown as the solid line axes) that are consistently aligned in the embedding space. In fact, (15) depicts the *coordinate transformation* between the landmarks' coordinates under the two bases. As a direct consequence, the corresponding *basis transformation* can be written as

$$\boldsymbol{W}_i^{\mathcal{E}} = \boldsymbol{W}_i \cdot \boldsymbol{T}_i = \left[\boldsymbol{q}_1^{(i)}, \boldsymbol{q}_2^{(i)}, \ldots, \boldsymbol{q}_d^{(i)}\right] \quad (i = 1, 2, \ldots, M). \tag{24}$$

Since $\boldsymbol{W}_i^{\mathcal{E}}$ directly describes the latent modes of variability of the high-dimensional data, we analogously call $\boldsymbol{q}_1^{(i)}, \boldsymbol{q}_2^{(i)}, \ldots, \boldsymbol{q}_d^{(i)}$ Latent Components (LCs), each component $\boldsymbol{q}_j^{(i)}$ $(j = 1, 2, \ldots, d)$ characterizing one axis of the embedding space. With (24), we then rewrite (18) as

$$\boldsymbol{x}_m^{(i)} - \bar{\boldsymbol{x}}^{(i)} = \boldsymbol{W}_i^{\mathcal{E}} \cdot \left(\boldsymbol{y}_m^{(i)} - \tilde{\boldsymbol{y}}_{L(1)}^{(i)}\right). \tag{25}$$

One can see that as the *factor loading matrix* in Factor Analysis (FA) [12], [14], the LCs plays a similar role in establishing a direct connection between the representations of manifold data in the high and low-dimensional spaces, thus it can be expected to find potential uses in many applications, e.g., manifold denoising, sample interpolation. In addition, some previous alignment methods like [41], [45], have used FA to fit their local models and finally also resulted in a parametric mapping. While the operation to translate global latent coordinates into directions in the input space also applies to these methods, they have paid less attention to this issue and not given an explicit modeling of the hidden variation modes.

4.4 Discussion

4.4.1 Comparisons with Previous Work

It can be seen that MLE bears some resemblance to global coordination [35] and subsequent methods [6], [41], [44], [53], [54]. Generally speaking, these methods all share the similar philosophy of aligning local linear models in a global coordinate space, which is first proposed in [35].

Both [35] and [44] use expectation-maximization (EM) to fit and align local linear models. This makes the algorithms quite inefficient, though [44] improves the training algorithm of [35] for a more constrained model. Moreover, as indicated in [35], because such EM-based methods are susceptible to local optima, they need a good initialization based on other methods (e.g., LLE or ISOMAP) to supervise the iterative optimization procedure.

Differently from [35], [44], the charting [6], LLC [41] and our MLE can all be viewed as post coordination, where the local models are coordinated or aligned after they have been fit to data [41]. By decoupling the local model fitting and coordination phases, all three methods produce closed-form solutions and gain efficiency in a noniterative scheme. Based on convex cost functions, they effectively avoid local optima in the coordination phase. However, the charting method builds one local model for each point, so its scaling is the same as that of LLE and ISOMAP, which is computationally demanding [29]. In contrast, LLC and our MLE only need to solve an eigenproblem scaling with the number of local models, which is far less than the number of training points.

We further compare MLE with LLC [41]. While LLC mainly serves as a general alignment method, the work presented in [41] has exploited a mixture of factor analyzers (MFA) [14] in the first phase, i.e., local model fitting. The construction of MFA is performed using an EM algorithm, which is likely to get stuck in local minima and be hampered by the presence of outliers, as indicated in [30]. Furthermore, LLC requires careful optimization of the number of local models in addition to the optimization of the parameters of the local models. The proposed MLP, though not guaranteed to be the optimal local linear models, has an explicit measure of the nonlinearity degree, which thereby facilitates the determination of the proper number of local models. In the second phase, i.e., the coordination, both methods need to solve the linear transformation (denoted by $\boldsymbol{L}_i$) from each local model to the global embedding space. LLC incorporates the parameter $\boldsymbol{L}_i$ into the LLE cost function and then directly obtains $\boldsymbol{L}_i$ by solving an eigenproblem, which requires the intrinsic dimensionality d to be specified a priori. The unit covariance constraint imposed by LLC will also lead to undesired rescaling of the manifold. On the contrary, the LGA algorithm in MLE can be considered as to first pursue the global space explicitly by exploiting the similar convex cost function as ISOMAP, and then solve $\boldsymbol{L}_i$ in a spectral regression way. This procedure not only gives rise to an automatic dimensionality estimation method in Section 4.3.4, but also enables us to preserve both global shape information and local structure more faithfully.

In addition, the local tangent space alignment (LTSA) [54] and locally multidimensional scaling (LMDS) [53] both share the similar alignment method to LLC in spirit. The local models in both LTSA and LMDS are still k-NN neighborhood, which is very crucial to the success of the methods, as pointed in [53] and [54]. Like charting [6], LTSA builds extremely overlapping local models on each data point. To alleviate this heavy redundance, LMDS seeks to find an approximate minimum set of the overlapping neighborhoods. Moreover, both methods do not derive a parametric mapping function. Although LMDS addresses a nonparametric out-of-sample extension, it suffers from the same computational cost problem as [4], and no further experimental justification is provided in [53].

4.4.2 Complexity Analysis

Basically, the computational complexity of MLE is dominated by the following four parts.

1. Computing the three $N \times N$ matrices $\boldsymbol{D}_E$, $\boldsymbol{D}_G$, and $\boldsymbol{R}$. The complexity of $\boldsymbol{D}_E$ computation is $O(N^2)$. $\boldsymbol{D}_G$ can be computed using Dijkstra's algorithm with Fibonacci heaps in $O(N^2 \log N + kN^2/2)$ time (k is the neighborhood size in the k-NN graph) [29]. $\boldsymbol{R}$ is computed in $O(N^2)$.
2. Constructing MLPs based on *Algorithm 1*. From Table 1, one can see that most steps of the algorithm are accessing operations against existing matrices computed in advance. The major computation is in step 3.3 to compute the nonlinearity score $S^{(i)}$ for each MLP according to (2). For simplicity, we assume the two child MLPs, $\boldsymbol{X}_l^{(i)}$ and $\boldsymbol{X}_r^{(i)}$, are of equal size. Thus, the total complexity of *Algorithm 1* is
$$O\left(\sum_{p=1}^{\lfloor \log N \rfloor} (2^p (N/2^p)^2)\right) \approx O(N^2).$$
3. Building local PCA models. For each MLP $\boldsymbol{X}^{(i)}$ with its data matrix of size $D \times N_i$, the PCA mainly involves eigenvalue decomposition of the $D \times D$ covariance matrix. Since it is often the case that in real problems $D \gg N_i$, the eigendecomposition can be conducted on a $N_i \times N_i$ matrix plus some additional matrix multiplications whose complexity can be ignored. Thus, the time complexity of this step is $O(\min(D, N_i)^3)$ for each of the M MLPs.
4. Aligning the local models by MDS. As discussed above, MDS is applied to a set of landmarks, whose minimal number for each model is $d + 1$. So the complexity of MDS is $O(M^3 d^3)$, which scales mainly with the number of local models M. This exhibits significant efficiency compared with $O(N^3)$ in ISOMAP and LLE. Once MDS is finished, the remaining computations to align the local models only involve several matrix multiplications in (16), (17). Note that the matrix inverse in (16) is only $d \times d$, which can be conducted very efficiently.

To sum up, the total complexity of MLE is the sum of the above four parts, which can be approximated by
$$O\left(N^2 \log N + kN^2 + \sum_{i=1}^{M} \min(D, N_i)^3 + M^3 d^3\right).$$
Generally, N_i, M, and d are far smaller than N, hence the complexity is roughly $O(N^2 \log N)$, i.e., the complexity in the first part is a major burden.

5 Experimental Results

In this section, extensive experiments on both synthetic and real data are conducted to validate the proposed MLE for dimension reduction and data reconstruction.

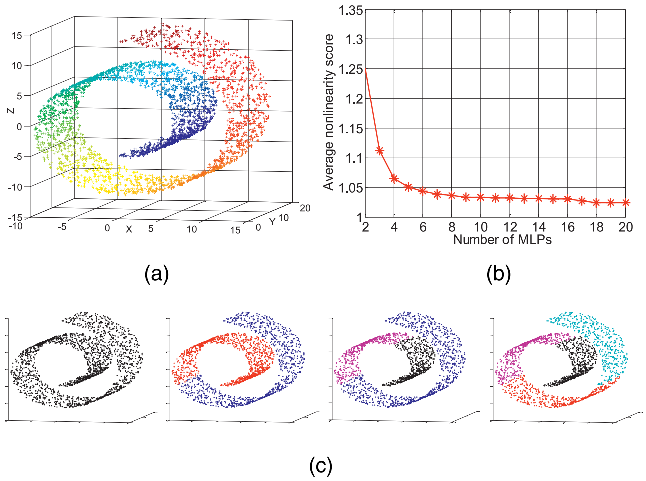


Fig. 6. Applying HDC to the "swiss-roll". (a) Original sampled data. (b) The average nonlinearity score curve. (c) The first four levels clustering dendrogram. MLPs are encoded with varying colors.

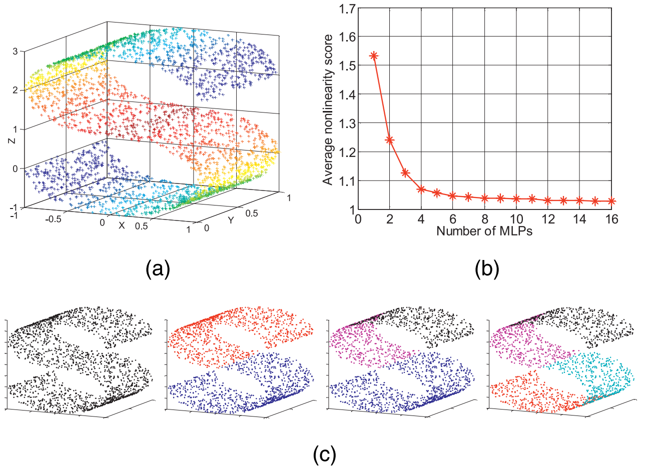


Fig. 7. Applying HDC to the "s-curve." Figures in (a)-(c) are similar to those in Fig. 6.

5.1 Experiments on Synthetic 3D Data

First, we illustrate the algorithm on two benchmark synthetic data sets: the "swiss-roll" and "s-curve." For each set, 3,000 points were randomly sampled from the original 3D manifold surface. The parameters in MLE include: 1) the neighborhood size, k; 2) the number of hard partitioning MLPs, P; and 3) the number of landmarks in each MLP, n_i. They were tuned in the same manner for both sets. Note that as stated in Section 4.1.2, the final number of MLPs after soft partitioning is denoted by M.

By specifying $k = 12$, *Algorithm 1* was first applied to compute the hard partitioning MLPs. The HDC results for both sets are shown in Figs. 6 and 7. In the following experiments, according to the average nonlinearity score curves, we chose the typical value of P as 20 and 16 for the two data sets, respectively, and selected about 10 percent of the training data as landmarks.

For a systematic empirical evaluation, we compared our MLE with three classical methods: ISOMAP, LLE, and LLC. Since LLC shares the similar two-phase procedure (i.e., local model fitting + coordination) with MLE, to further investigate their differences we implemented a variant of MLE, called MLP Coordination (MLPC). The variant simply takes our MLP-based PCA subspaces as local models in the first phase, but uses the alignment method of LLC instead of our LGA in the second phase.

To conduct quantitative comparison between different algorithms, we assess the quality of the resulting low-dimensional embeddings by evaluating to what extent the global and local structure of the data is retained. The evaluation is performed in two ways: 1) by measuring the embedding error (as is done in [45]) and 2) by measuring the trustworthiness and the continuity errors of the embeddings (as is used in [30] and [56]). The embedding error measures the squared distance from the recovered low-dimensional embedding to the known true latent coordinates. Due to the unit covariance constraint in LLE, LLC, and MLPC, the global metric information will be lost in these methods. To enable their comparison with MLE and ISOMAP, we simply scaled the true 2D latent coordinates to $[-1, 1]$, as shown in the top row of Fig. 8, and optimally linearly transform the recovered embeddings of different methods to the true latent coordinates as in [45]. The *embedding error* is then defined as follows:

$$E = \sqrt{\sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{y}_n^*\|^2}, \tag{26}$$

where N is the sample number, $\boldsymbol{y}_n$ and $\boldsymbol{y}_n^*$ represent the recovered and true latent coordinates, respectively. It is easy to see that the embedding error tends to measure the global structure distortion of the manifold. To measure the local structure distortion, we resort to the trustworthiness and continuity errors. The *trustworthiness error* measures the proportion of points that are too close together in the low-dimensional space, and is defined as

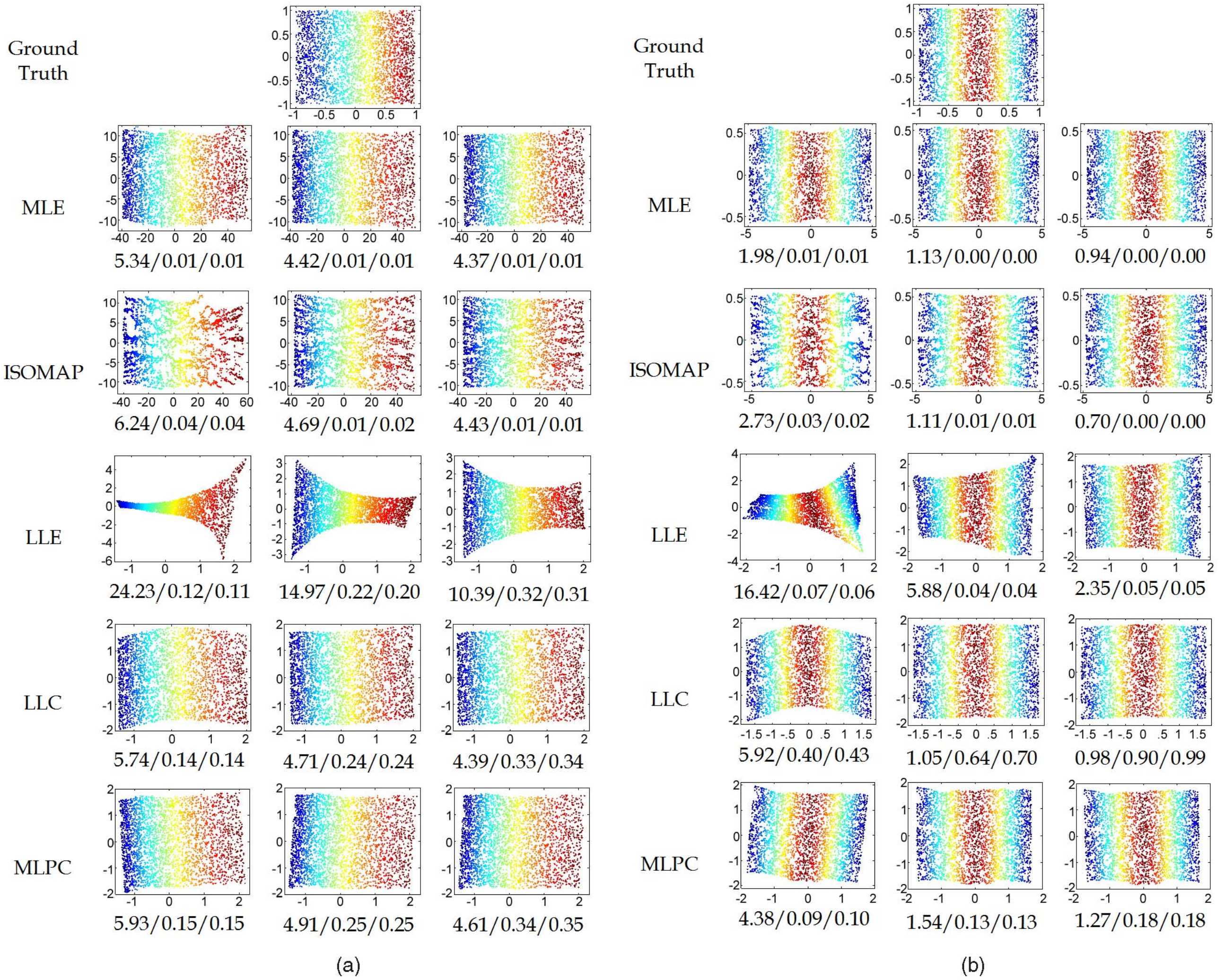
$$T(k) = 100 \times \frac{2}{Nk(2N - 3k - 1)} \sum_{n=1}^{N} \sum_{m \in U_n^{(k)}} (r(n, m) - k), \tag{27}$$

where k is the neighborhood size, $r(n, m)$ is the rank of the point $\boldsymbol{x}_m$ in the ordering according to the pair-wise distance from point $\boldsymbol{x}_n$ in the high-dimensional space. The variable $U_n^{(k)}$ denotes the set of points that are among the k-NNs of $\boldsymbol{y}_n$ in the low-dimensional space but not in the high-dimensional space. In contrast, the *continuity error* measures the proportion of points that are pushed away from their neighborhood in the low-dimensional space, and is analogously defined as

$$C(k) = 100 \times \frac{2}{Nk(2N - 3k - 1)} \sum_{n=1}^{N} \sum_{m \in V_n^{(k)}} (\hat{r}(n, m) - k), \tag{28}$$

where $\hat{r}(n, m)$ is the rank of the point $\boldsymbol{y}_m$ in the ordering according to the pair-wise distance from point $\boldsymbol{y}_n$ in the low-dimensional space. The variable $V_n^{(k)}$ denotes the set of points that are among the k-NNs of $\boldsymbol{x}_n$ in the high-dimensional space but not in the low-dimensional space. In the following Figs. 8, 9, and 10, the three errors are written under each embedding and in the form of "Embedding/Trustworthiness/Continuity" (abbreviated as **E./T./C.**).

Experiment 1: Influence of k. To evaluate the robustness to varying neighborhood size k, we have tried sizes from 6 to 18 points and compare results of different methods in Fig. 8.



(a) (b)

Fig. 8. Comparison of different algorithms with varying neighborhood size k on two synthetic data sets, (a) swiss-roll and (b) s-curve. Results in the three columns correspond to $k =$ 6, 12, and 18, respectively. The values under each embedding give the error measures in the form of "*Embedding*/*Trustworthiness*/*Continuity*" (**E./T./C.**).

In our comparisons, we show the best LLC result among several trials for each parameter, since multiple runs of the algorithm will initialize different MFA local models, thus yielding different results. The number of local models used in LLC and MLPC is 50 and 30, respectively. From the figure, it is confirmed that the proposed MLE, as ISOMAP, has preserved the global metric information and produced more faithful embeddings. In contrast, the aspect ratio is mostly lost in LLE, LLC, and MLPC due to their unit covariance constraint. As a local approach, LLE is the most sensitive to k on preserving global shape information of the manifold. LLC and MLPC are also shown to generate some deformations especially under smaller neighbor size. The advantage of MLE over other methods can be more clearly demonstrated when observing the quantitative error measures in the figure. Specifically, while LLC delivers comparable E.-error to MLE, its T./C.-error is significantly larger than that of MLE. A similar phenomenon can also be observed from the comparison between MLE and MLPC, which only differ in their alignment methods for global coordination. Such results verify that MLE can show more reliability on preserving local geometry. We believe that the success of MLE is attributed to both its efficient local model MLP and its global coordination method LGA. In addition, our experiment shows that when applied to evaluate the embedding, the E.-error and the T./C.-error measures complement each other since a low E.-error measure does not necessarily imply the similar low T./C.-error measure.

Experiment 2: Influence of P. Here the number of local linear models P is a direct reflection of the threshold parameter δ in Algorithm 1, as noted in the end of Section 4.1.1. Intuitively, the parameter P plays a trade-off between computational cost and representation accuracy. That is, a smaller P implies fewer MLPs (thus more efficiency) but larger linearity deviation within each MLP, and vice versa. Take the above "swiss-roll" data for example. According to Fig. 6b, in MLE and MLPC, we have tested different values of P from 5 to 30 MLPs. For fair comparison with LLC/MLPC, we used only the hard partitioning MLPs for subsequent LGA procedure of MLE in this experiment. For LLC, we also tried different numbers of local models under the same neighbor size $k = 12$ as MLE. Fig. 9 gives the results from the three methods along with respective computation time. As expected, MLE yields more and more stable results with increased local models. Even with very few MLPs, say $P = 5$, it can still output

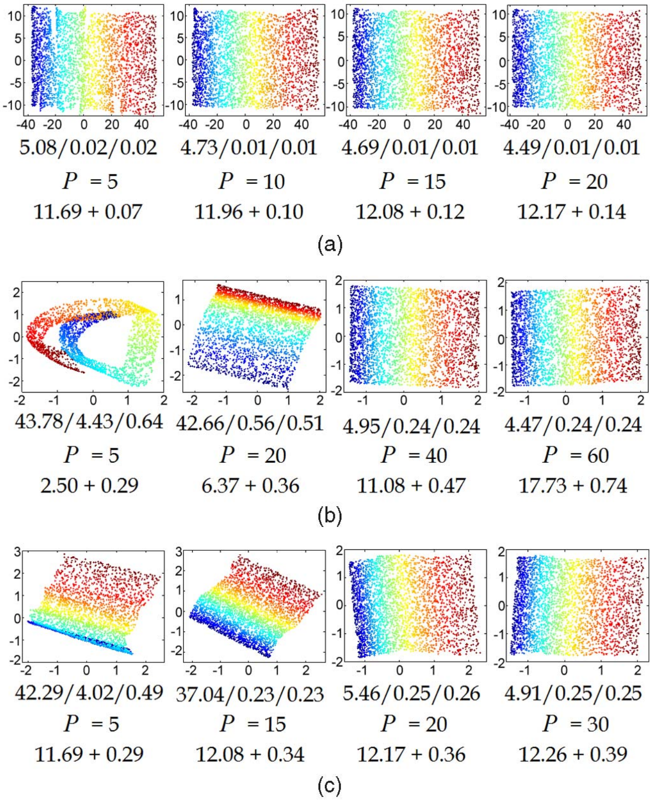


Fig. 9. Comparison of (a) MLE, (b) LLC, and (c) MLPC with different numbers of local models. The three rows under each embedding are: the E./T./C. error measures, the number of local models, and the computation time (*seconds*). The time is given in the form of "*the first phase* (fitting)" + "*the second phase* (coordination)."

desirable embedding. On the contrary, LLC is shown to be more sensitive to the setting of P, and it requires more local models (i.e., MFA) to unfold the curved data reliably. When combining our local models (i.e., MLP) with the alignment procedure of LLC, the variant MLPC shows improved embeddings over LLC; however, like LLC, it produces substantial deformation with small numbers of local models. Also note that both LLC and MLPC have much larger T./C.-error than MLE, even though they are under the similar E.-error. These comparisons, in one aspect, again verify the *economical* and *efficient* merit of MLP, and in another aspect, demonstrate the advantage of LGA over LLC for aligning the local models. In terms of the computation time, both MLE and LLC spend the most part in local model fitting. We observe that with the same increase of local models, e.g., from 5 to 20, the time cost increase for MLE is much less than that for LLC. The reason is that, as discussed in Section 4.4.2, the major burden of MLE lies in the computation of geodesic distances, while the HDC algorithm only takes very little time. In LLC, however, the time grows proportionally to the number of local models, and each local model is iteratively optimized to a factor analyzer by an EM algorithm.

Experiment 3: Influence of n_i. For the "swiss-roll" data, under $P = 20$, finally $M = 57$ MLPs were discovered after the soft partitioning. As stated in Section 4.3.1, the number of landmarks in each MLP should satisfy $n_i \geq d + 1$, where the intrinsic dimensionality here is $d = 2$. To investigate its effect on MLE, by specifying different values, we pursued the 2D embeddings and computed the residual variances as [42]. In Fig. 10, more stable embedding with decreased E./T./C. errors and residual variance can be yielded as the landmarks increase. Even relying on the least number $(d + 1)$ of landmarks, a favorable result with slight distortion can still be obtained. When $n_i = 5$, i.e., a total of $57 \times 5 = 285$ landmarks (about 10 percent of the training set) were used, the residual variance gets comparable to that of ISOMAP at 5×10^{-4}. However, in this case, ISOMAP confronts a much larger eigenproblem of size $3{,}000 \times 3{,}000$, compared with 285×285 in MLE.

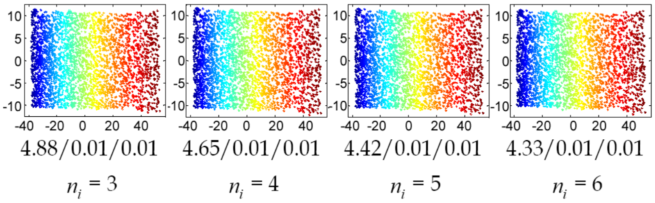


Fig. 10. MLE 2D embeddings and corresponding E./T./C. error measures with different numbers of landmarks.

In addition to testing different parameters, we next highlight several theoretical issues of MLE through empirical observations on the "swiss-roll" manifold.

1. *Orthogonality of transition matrix* $\boldsymbol{T}_i$. For each of the 57 local models, we compute the Frobenius-norm of the matrix $(\boldsymbol{T}_i)^T\boldsymbol{T}_i - \boldsymbol{I}$, where $\boldsymbol{T}_i$ is the local transition matrix in (15) and $\boldsymbol{I}$ is the identity matrix. From Fig. 11a, we see that most values are very close to the target value 0.
2. *Estimation of intrinsic dimensionality* d. Under the correct estimation $d^* = 2$, we observe the transformation error, i.e., each summed term in (23). As shown in Fig. 11b, the errors are indeed very small when considering the magnitude of the MLE embedding space in Fig. 8a. Over a total of 285 landmarks, the mean error 0.247 is even smaller than the mean nearest neighbor distance 0.389 that is computed among all data points in the manifold.
3. *Validity of the Latent Component*. In Fig. 6a, the first variation mode of "swiss-roll" is along the twisting direction in the XOZ plane and the second one is along the depth direction parallel to the Y-axis. While different local models twist along varying direction vectors, they all share a common depth direction vector of $[0, 1, 0]^T$ in the observation space. As in (24), we thus compare Latent Component $\boldsymbol{q}_2^{(i)}$ $(i = 1, 2, \ldots, 57)$ with the vector $[0, 1, 0]^T$ and demonstrate their correlation coefficient for each local model in Fig. 11c. The result turns out to be that the Latent Component is almost perfectly high-correlated with the essential variation mode. This observation supports that our algorithm is able to explicitly model the underlying variations of the manifold.

5.2 Experiments on Synthetic Image Data

To validate MLE on high-dimensional data, we first used the ISOFace data set [42], which consists of 698 synthetic face images of $64 \times 64 = 4{,}096$ pixels each. All faces lie on

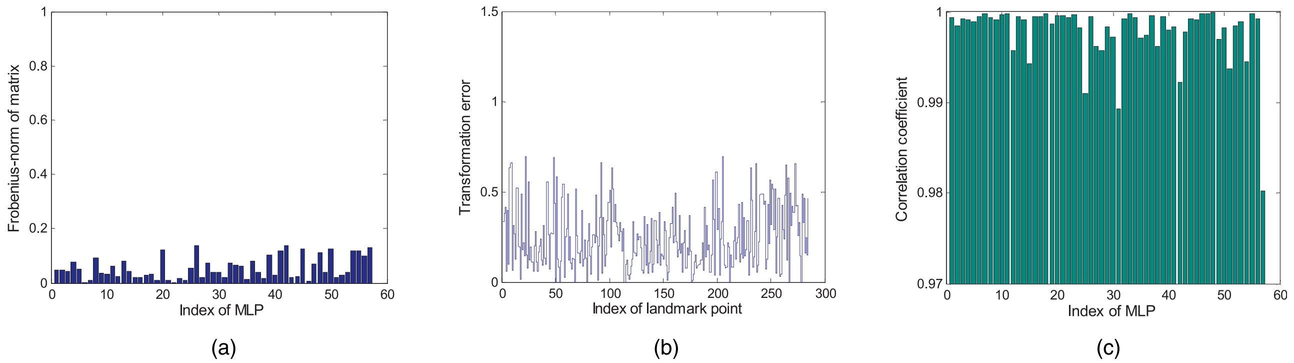


Fig. 11. Evaluation of three theoretical issues of MLE on the "swiss-roll" manifold. (a) Frobenius-norm of the matrix $(\boldsymbol{T}_i)^T\boldsymbol{T}_i - \boldsymbol{I}$. (b) Transformation error for the landmarks. (c) Correlations between the Latent Component and the direction vector of the variation mode.

an intrinsically 3D manifold parameterized by two pose variables plus an azimuthal lighting angle [42]. The whole set was divided into a training set with the first 650 images, and a test set with the remaining 48 ones. Note that in the original set, images are randomly ordered.

Qualitative evaluation. We compared with ISOMAP to evaluate how well MLE can perform to unravel very high-dimensional raw data and further to yield parametric mapping. To learn the manifold, ISOMAP used all 698 samples and MLE employed only the 650 training images both with setting $k = 6$ as [42]. For MLE, $M = 27$ MLPs were finally discovered. By specifying $n_i = 7$, we thus used a total of $27 \times 7 = 189$ landmarks, about 30 percent of the training data. Both methods have correctly discovered the 3D face manifold, with the first 2D embeddings visualized in Fig. 12. One can see again that, similarly to ISOMAP, our MLE has preserved the underlying global structure of the manifold whereas it used a relatively smaller training set.

After manifold learning, MLE then allows for out-of-sample extensions by parametric mapping. We first applied forward mapping to the test data (index from 651 to 698) to appropriately locate them in the reduced dimensional space. Fig. 12 also shows several examples, with each image denoted by its index. As can be seen, these testing samples successfully find their coordinates which reflect their intrinsic properties, i.e., left-right and up-down pose. We then synthesized a series of virtual views as shown in Fig. 13 by the backward mapping. There may be question that some virtual faces seem not as good as the raw images. Two reasons may be adduced: One is the sparseness of the training set; the other is that each face is reconstructed by only three components (since $d = 3$).

Quantitative comparison. We made further comparisons between the generalization performance of MLE and LLC/MLPC in terms of reconstruction error as [45]. As MLE, both LLC and MLPC also used the 650 training images to learn the parametric mapping with $P = 30$ and 27 local models, respectively, under the setting of $k = 6$ neighbors. The learned mappings from all the three methods were then utilized to reconstruct each sample in the test set. For each test sample $\boldsymbol{x}_n$, its reconstruction $\hat{\boldsymbol{x}}_n$ is obtained by mapping $\boldsymbol{x}_n$ to a single point $\boldsymbol{y}_n$ in the embedding space and then mapping $\boldsymbol{y}_n$ back to the image data space [45]. The reconstruction error is defined as

$$E_n = \frac{1}{\sqrt{D}} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|, \tag{29}$$

where D (in this case $D = 4{,}096$) is the dimension of the image space. Intuitively, the error (29) measures the average perturbation over all pixels in the test image. Note that, each pixel is quantized to $[0, 255]$ in our experiment.

Since the alignment procedure of LLC requires the latent dimensionality d to be specified a priori, we have tried different values of d ranging from 3 to 20 for LLC and MLPC. The errors are summarized in Table 4 and some of the reconstructions are shown in Fig. 14, where "MLE_3" depicts MLE trained with $d = 3$ and the others have analogous meanings. The reported results in Table 4 are averages and standard deviations over the 48 test samples. We find that MLE, with $d = 3$, can perform as well as LLC with a much higher $d = 20$; while LLC fails to reconstruct the face images with the intrinsic dimensionality well ($d = 3$). While MLPC outperforms LLC with decreased reconstruction error under the same dimensionality as the findings in the "swiss-roll" data, it still exhibits considerably inferior performance

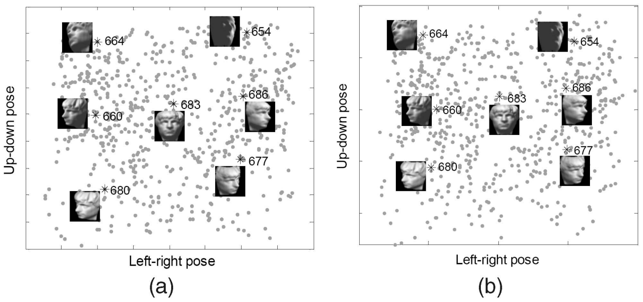


Fig. 12. 2D embeddings of the ISOFace manifold discovered by (a) ISOMAP and (b) MLE, respectively. Stars in the figure denote test samples with corresponding images and indices superimposed. Note that, in the ISOMAP result (a), "test" images are coprojected with training samples by the ISOMAP training procedure.

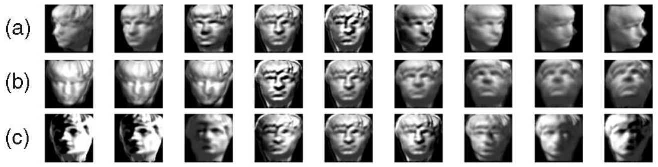


Fig. 13. Each row contains faces reconstructed from test points along an axis-parallel line in the 3D embedding space. From top to bottom: left-right pose, up-down pose, and light direction variations.

TABLE 4
Reconstruction Errors for the ISOFace Data Set

	MLE_3	LLC_3	LLC_10	LLC_20
Error	18.9 ± 7.3	47.7 ± 20.1	35.1 ± 18.0	22.2 ± 6.8
	MLPC_3	MLPC_6	MLPC_10	MLPC_20
Error	45.9 ± 17.8	28.9 ± 11.9	23.7 ± 9.5	19.0 ± 7.2

compared to MLE. We attribute the gain of MLE to the high accuracy of its MLP-based PCA modeling and the cost controllable LGA coordination facilitated by (23). It is also worth noting that, given different values of parameter d, LLC needs to run repeatedly to solve different eigenproblems of varying sizes. The training time here for MLE_3 and LLC_3/10/20 is 11.5 and 27.6/35.0/49.5, respectively, all in seconds. We find that the MFA fitting in LLC on high-dimensional data is quite time demanding from this experiment.

Estimation of intrinsic dimensionality. To check the feasibility of the method described in Section 4.3.4, here we show the intermediate results on this synthetic data set.

Figs. 15a and 15b show the estimations from PCA and MDS, respectively, where the PCA energy preserving ratio under each dimension was computed by averaging the ratios from all the 27 local models. Within a roughly estimated interval $[2, 10]$ for possible d, according to (23), we computed the total transformation error of the 189 landmarks for each value in this interval. On first sight it seems that higher dimensionality would always lead to smaller error and the criterion in (23) would thus favor a large value for d. However, it should also be noted that once d exceeds its proper value, the added higher dimensional coordinates in $\boldsymbol{z}^{(i)}_{L(k)}$ and $\hat{\boldsymbol{y}}^{(i)}_{L(k)}$ (both are d-dimensional vectors in (23)) will also inevitably cause increase in the transformation error. Therefore, the cost function (23) does not always decrease with increasing the dimensionality. The experimental result in Fig. 15c verifies the above analysis. The correct estimation of $d = 3$ for ISOFace data demonstrates the potential of our method to be applied to other more complex high-dimensional manifold.

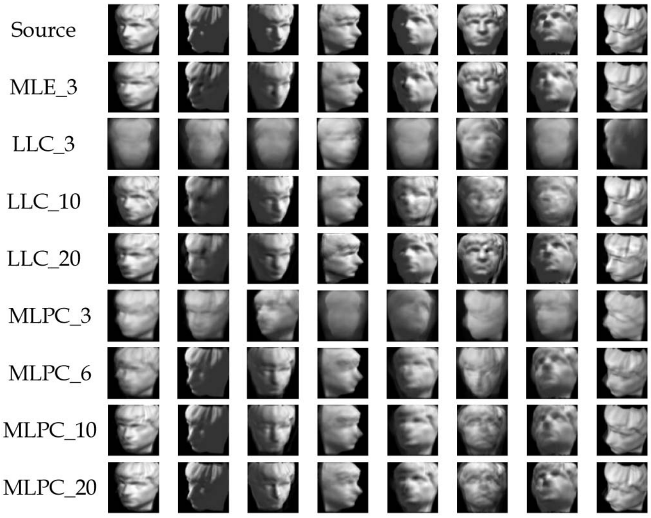


Fig. 14. Some source face images and corresponding reconstructions from MLE/LLC/MLPC under different latent dimensionalities.

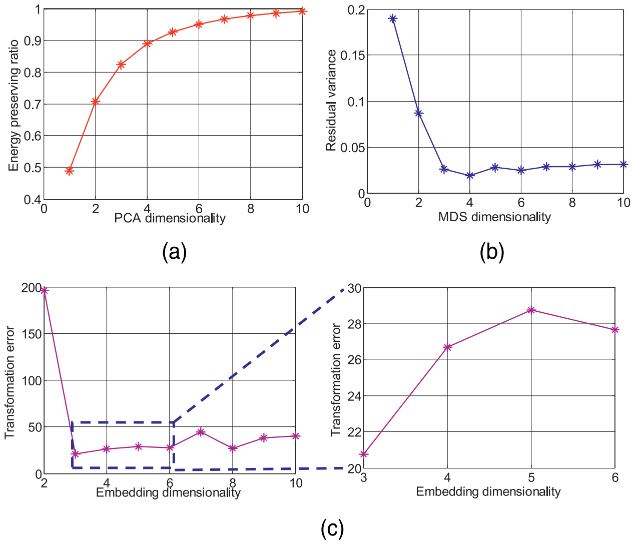


Fig. 15. Estimation of intrinsic dimensionality on the ISOFace manifold. (a) and (b) give the estimation from PCA and MDS, respectively. (c) shows the estimation using the proposed cost function in (23).

5.3 Experiments on Realistic Video Data

In this section, we test MLE on another data set, called LLEFace [34], which contains real faces believed to reside on a complex manifold with few degrees of freedom. The 20×28 face images come from a 1,965-frame video [34] in which a single person strikes a variety of poses and expressions, along with heavy synthetic camera jitters. The data set has also been widely used in [35], [41], [45], etc.

Qualitative evaluation. We first applied MLE on the whole 1,965 samples. For comparison with LLC, the same parameter setting as [41] was used. With $k = 36$ neighbors, we chose $P = 10$ by HDC. After soft partitioning, $M = 26$ MLPs were constructed at last. Since the true latent dimension of this real image set is not known, we set $d = 8$ as [41]. By specifying $n_i = 15$, in total $26 \times 15 = 390$ landmarks (about 20 percent of the training data) were then exploited to map the face images from 560D image space to an 8D embedding space. Fig. 16 illustrates the first 2D embedding and some reconstructions. Similarly to previous work [35], [41], [45], the 2D MLE embedding correctly discovers the two dominant variations in the face manifold, one for pose and another for expression. One may also see that some reconstructions near the boundary are not good enough. This is mainly because the model is extrapolating from the training images to low sample density regions.

Further discussion on the Latent Component. As discussed in Section 4.3.4, those virtual faces in Figs. 13 and 16 are in fact reconstructed along the directions of Latent Components via (25). In analogy to Eigenface in the face recognition literature, we call the Latent Component here as Latentface. Fig. 17 shows the Eigenfaces and Latentfaces from one local model of the LLEFace manifold. While Eigenfaces describe the directions with the largest variances in the high-dimensional data space, Latentfaces describe the directions which dominate the intrinsic (latent) variability

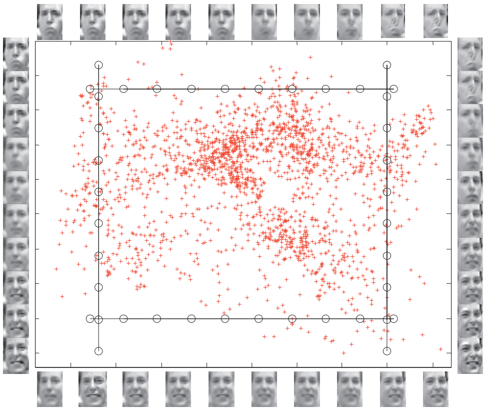


Fig. 16. The first 2D embedding discovered by MLE for the LLEFace manifold. *Red pluses* indicate the global coordinate for each training example. The images on the borders are reconstructed from the global coordinates specified by the corresponding *open circles* (samples along the straight lines in the global space).

of the manifold. Therefore, when we reconstruct face images along the directions of Latentfaces, they will exhibit the intrinsic modes of variability, which correspond to the global data variations. With this merit, Latentfaces can be expected to find potential widespread applications in various problems, such as pose estimation, facial expression analysis, face recognition, and so on.

Quantitative comparison. As in the ISOFace data set, we compared our MLE against LLC to show how their performances depend on the amount of training data and the number of local models. From the total 1,965 samples, we used varying percentage of training data (ranging from 60 to 90 percent) to learn the mapping and the rest as test data to assess the reconstruction quality. We measure the average reconstruction error over all test samples $\boldsymbol{x}_n$:

$$E_{rec} = \frac{1}{N\sqrt{D}} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|. \tag{30}$$

Similarly to (29), here $D = 560$ is the dimension of the image space, and N is the number of test samples, which takes 785, 589, 393, and 196, respectively, for each train percentage.

We trained MLE and LLC using $k = 36$ and $d = 8$, while varying the number P of local models (ranging from 10 to 25). Again, for fair comparison, we used only the hard partitioning MLPs for MLE. We tabulated the results in Table 5, where each error is an average and standard deviation over five randomly drawn train and test sets for each percentage. The results show that MLE is always able to deliver higher accuracy than LLC and both methods generally obtain decreased errors with more training data, as expected. Moreover, as the number P grows larger, the errors of MLE consistently become smaller thanks to the increased accuracy with more MLPs. In comparison, LLC shows moderate overfitting when more parameters need to be estimated for many local models, as also found in [45].

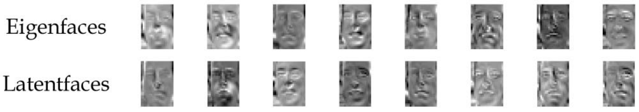


Fig. 17. Example of Eigenfaces and Latentfaces corresponding to one local model. We used the latent dimensionality of $d = 8$.

TABLE 5
Reconstruction Errors for the LLEFace Data Set

P	10	15	20	25
60%	12.09 ± 0.29 13.15 ± 0.28	11.53 ± 0.31 12.93 ± 0.06	11.17 ± 0.17 13.20 ± 0.36	10.83 ± 0.14 13.32 ± 0.44
70%	11.91 ± 0.28 12.99 ± 0.05	11.48 ± 0.28 12.82 ± 0.40	11.06 ± 0.29 12.83 ± 0.14	10.77 ± 0.28 13.22 ± 0.12
80%	11.82 ± 0.17 12.87 ± 0.27	11.31 ± 0.26 12.79 ± 0.46	10.94 ± 0.32 12.74 ± 0.42	10.78 ± 0.27 13.18 ± 0.05
90%	11.76 ± 0.11 12.76 ± 0.54	11.12 ± 0.03 12.78 ± 0.28	10.80 ± 0.18 13.13 ± 0.37	10.54 ± 0.17 13.89 ± 0.77

For each percentage, MLE results are printed above LLC results.

6 Conclusion and Future Work

We propose a manifold learning method, Maximal Linear Embedding. Compared to classic ISOMAP and LLE, our approach can well preserve both local geometry and global structure of the manifold. The method further derives a parametric function for out-of-sample extension. Unlike the locally linear neighborhood in LLE, MLE defines maximal linear patch as the basis for linear embedding, which is more reasonable and efficient. Since MLP is constructed according to the geodesic distance, our method also exploits the most essential point of ISOMAP. In comparison with related parametric methods such as LLC, MLE improves upon them in both phases of local model fitting and coordination, as discussed in Section 4.4.1. Experimental results in Section 5 indicate that MLE compares favorably to LLC in the sense that fewer local models are required to pursue reliable low-dimensional embedding, and smaller reconstruction errors can be obtained under the similar parameter settings.

One interesting research direction is to introduce a probabilistic model into our MLP, as in LLC and CFA, which will give a notion of uncertainty in the mapping and result in more stability and flexibility. Currently, our coordination method LGA exploits the similar rigid constraint of isometry as ISOMAP, which might limit their applications. Inspired by [29], [57], we will investigate a more flexible algorithm to achieve a trade-off between the rigid constraint of isometry and the deficiency of global metrics. Moreover, we will make an effort to study two issues plaguing almost all manifold learning methods, noise sensitivity and sampling density, to extend our work to more practical and challenging applications.

Acknowledgments

R. Wang was with the Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, P.R. China, where most of this work was done when he was a PhD candidate. The authors would like to thank

Dr. Y. Ma and the anonymous reviewers for their detailed comments and constructive suggestions. They are also grateful to Dr. Yee Whye Teh for sharing the code of LLC. This work was partially supported by the Natural Science Foundation of China under contract, No. 61025010 and No. 60872077, and the National Basic Research Program of China (973 Program) under contract 2009CB320902.

REFERENCES

[1] M. Balasubramanian and E.L. Schwartz, "The IsoMap Algorithm and Topological Stability," *Science,* vol. 295, no. 4, p. 5552, Jan. 2002.

[2] P. Baldi and K. Hornik, "Neural Networks and Principal Component Analysis: Learning from Examples without Local Minima," *Neural Networks,* vol. 2, pp. 53-58, 1989.

[3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems,* vol. 14, pp. 585-591, 2002.

[4] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-of-Sample Extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems,* vol. 16, pp. 2197-2219, 2004.

[5] C.M. Bishop, M. Svensen, and C.K.I. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation,* vol. 10, pp. 215-234, 1998.

[6] M. Brand, "Charting a Manifold," *Advances in Neural Information Processing Systems,* vol. 15, pp. 961-968, 2003.

[7] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local Discriminant Embedding and Its Variants," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 846-853, 2005.

[8] T.F. Cox and M.A.A. Cox, *Multidimensional Scaling.* Chapman and Hall, 2001.

[9] D. DeMers and G. Cottrell, "Nonlinear Dimensionality Reduction," *Advances in Neural Information Processing Systems,* vol. 5, pp. 580-587, 1993.

[10] D.L. Donoho and C. Grimes, "Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data," *Proc. Nat'l Academy of Sciences,* vol. 100, no. 10, pp. 5591-5596, May 2003.

[11] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* Wiley, 2000.

[12] B.S. Everitt, *An Introduction to Latent Variable Models.* Chapman and Hall, 1984.

[13] K. Fukunaga and D.R. Olsen, "An Algorithm for Finding Intrinsic Dimensionality of Data," *IEEE Trans. Computers,* vol. 20, no. 2, pp. 176-193, Feb. 1971.

[14] Z. Ghahramani and G.E. Hinton, "The EM Algorithm for Mixtures of Factor Analyzers," Technical Report CRG-TR-96-1, Univ. of Toronto, 1996.

[15] J. Ham, D. Lee, S. Mika, and B. Schölkopf, "A Kernel View of the Dimensionality Reduction of Manifolds," *Proc. Int'l Conf. Machine Learning,* pp. 47-54, 2004.

[16] T. Hastie and W. Stuetzle, "Principal Curves," *J. Am. Statistical Assoc.,* vol. 84, pp. 502-516, 1989.

[17] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood Preserving Embedding," *Proc. 10th IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 1208-1213, 2005.

[18] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 3, pp. 328-340, Mar. 2005.

[19] D.R. Hundley and M.J. Kirby, "Estimation of Topological Dimension," *Proc. SIAM Int'l Conf. Data Mining,* pp. 194-202, 2003.

[20] I.T. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1986.

[21] N. Kambhatla and T.K. Leen, "Dimension Reduction by Local Principal Component Analysis," *Neural Computation,* vol. 9, pp. 1493-1516, 1997.

[22] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1990.

[23] T. Kohonen, *Self-Organizing Maps,* third ed. Springer-Verlag, 2001.

[24] E. Kokiopoulou and Y. Saad, "Orthogonal Neighborhood Preserving Projections: A Projection-Based Dimensionality Reduction Technique," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2143-2156, Dec. 2007.

[25] S. Lafon and A.B. Lee, "Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. 1393-1403, Sept. 2006.

[26] M.C. Law and A.K. Jain, "Incremental Nonlinear Dimensionality Reduction by Manifold Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 3, pp. 377-391, Mar. 2006.

[27] E. Levina and P.J. Bickel, "Maximum Likelihood Estimation of Intrinsic Dimension," *Advances in Neural Information Processing Systems,* vol. 17, pp. 777-784, 2005.

[28] R.S. Lin, C.B. Liu, M.H. Yang, N. Ahuja, and S. Levinson, "Learning Nonlinear Manifolds from Time Series," *Proc. Ninth European Conf. Computer Vision,* vol. 2, pp. 245-256, May 2006.

[29] T. Lin and H. Zha, "Riemannian Manifold Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 5, pp. 796-809, May 2008.

[30] L. Maaten, E. Postma, and J. Herik, "Dimensionality Reduction: A Comparative Review," Technical Report TiCC-TR 2009-005, Tilburg Univ., 2009.

[31] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks,* vol. 12, no. 2, pp. 181-201, Mar. 2001.

[32] J. Park, Z. Zhang, H. Zha, and R. Kasturi, "Local Smoothing for Manifold Learning," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 452-459, 2004.

[33] K. Pettis, T. Bailey, A.K. Jain, and R. Dubes, "An Intrinsic Dimensionality Estimator from Near-Neighbor Information," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 1, pp. 25-36, Jan. 1979.

[34] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 22, pp. 2323-2326, Dec. 2000.

[35] S.T. Roweis, L.K. Saul, and G.E. Hinton, "Global Coordination of Local Linear Models," *Advances in Neural Information Processing Systems,* vol. 14, pp. 889-896, 2002.

[36] J.W. Sammon, "A Nonlinear Mapping for Data Structure Analysis," *IEEE Trans. Computers,* vol. 18, no. 5, pp. 401-409, May 1969.

[37] L.K. Saul and S.T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," *J. Machine Learning Research,* vol. 4, pp. 119-155, 2003.

[38] B. Schölkopf, A.J. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.

[39] H.S. Seung and D.D. Lee, "The Manifold Ways of Perception," *Science,* vol. 290, pp. 2268-2269, Dec. 2000.

[40] F. Sha and L.K. Saul, "Analysis and Extension of Spectral Methods for Nonlinear Dimensionality Reduction," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 785-792, 2005.

[41] Y.W. Teh and S.T. Roweis, "Automatic Alignment of Hidden Representations," *Advances in Neural Information Processing Systems,* vol. 15, pp. 841-848, 2003.

[42] J. Tenenbaum, V. Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 22, pp. 2319-2323, Dec. 2000.

[43] R. Tibshirani, "Principal Curves Revisited," *Statistics and Computing,* vol. 2, pp. 183-190, 1992.

[44] J. Verbeek, N. Vlassis, and B. Kröse, "Coordinating Principal Component Analyzers," *Proc. Int'l Conf. Artificial Neural Networks,* vol. 12, pp. 914-919, 2002.

[45] J. Verbeek, "Learning Nonlinear Image Manifolds by Global Alignment of Local Linear Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 8, pp. 1236-1250, Aug. 2006.

[46] R. Wang, S. Shan, X. Chen, and W. Gao, "Manifold-Manifold Distance with Application to Face Recognition Based on Image Set," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 2940-2947, 2008.

[47] R. Wang and X. Chen, "Manifold Discriminant Analysis," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 429-436, 2009.

[48] J. Wang, Z. Zhang, and H. Zha, "Adaptive Manifold Learning," *Advances in Neural Information Processing Systems,* vol. 17, pp. 1473-1480, 2005.

[49] K.Q. Weinberger and L.K. Saul, "Unsupervised Learning of Image Manifolds by Semidefinite Programming," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 988-995, 2004.

[50] G. Wen, L. Jiang, and N.R. Shadbolt, "Using Graph Algebra to Optimize Neighborhood for Isometric Mapping," *Proc. 20th Int'l Joint Conf. Artificial Intelligence,* pp. 2398-2403, 2007.

[51] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extension: A General Framework for Dimensionality Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 1, pp. 40-51, Jan. 2007.

[52] J. Yang, D. Zhang, J.Y. Yang, and B. Niu, "Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, pp. 650-664, Apr. 2007.

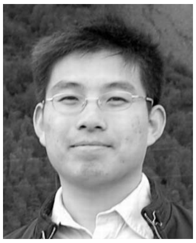
[53] L. Yang, "Alignment of Overlapping Locally Scaled Patches for Multidimensional Scaling and Dimensionality Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 3, pp. 438-450, Mar. 2008.

[54] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *SIAM J. Scientific Computing,* vol. 26, no. 1, pp. 313-338, 2004.

[55] D. Zhao and L. Yang, "Incremental Isometric Embedding of High-Dimensional Data Using Connected Neighborhood Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 1, pp. 86-98, Jan. 2009.

[56] J. Venna and S. Kaski, "Visualizing Gene Interaction Graphs with Local Multidimensional Scaling," *Proc. 14th European Symp. Artificial Neural Networks,* pp. 557-562, 2006.

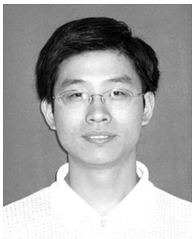
[57] V. de Silva and J.B. Tenenbaum, "Global versus Local Methods in Nonlinear Dimensionality Reduction," *Advances in Neural Information Processing Systems,* vol. 15, pp. 705-712, 2003.



Ruiping Wang received the BS degree in applied mathematics from Beijing Jiaotong University, Beijing, China, in 2003 and the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2010. Since July 2010, he has been a postdoctoral researcher in the Department of Automation, Tsinghua University, Beijing, China. He is currently working as a research associate with the Computer Vision Laboratory, Institute for Advanced Computer Studies (UMIACS), at the University of Maryland, College Park. He received the Best Student Poster Award Runner-up from IEEE CVPR 2008 for the work on Manifold-Manifold Distance. His research interests include computer vision, pattern recognition, and machine learning. He is a member of the IEEE.

Shiguang Shan received the MS degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2004. He has been with ICT, CAS since 2002 and has been a professor since 2010. He especially focuses on face recognition related research topics, and has published more than 100 papers on related research topics. He received China's State Scientific and Technological Progress Awards in 2005 for his work on face recognition technologies. One of his coauthored CVPR '08 papers won the "Best Student Poster Award Runner-up." He also won the Silver Medal "Scopus Future Star of Science Award" in 2009. His research interests include image analysis, pattern recognition, and computer vision. He is a member of the IEEE.

Xilin Chen received the BS, MS, and PhD degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively. He was a professor with the Harbin Institute of Technology from 1999 to 2005. He was a visiting scholar with Carnegie Mellon University, Pittsburgh, Pennsylvania, from 2001 to 2004. He has been a professor with the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, since August 2004. He is the director of the Key Laboratory of Intelligent Information Processing, CAS. He has published one book and more than 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces. He is an associate editor of the *IEEE Transactions on Image Processing*, an area editor of the *Journal of Computer Science and Technology*, and an associate editor of the *Chinese Journal of Computers*. He has served as a program committee member for more than 30 international conferences. He has received several awards, including China's State Scientific and Technological Progress Award in 2000, 2003, and 2005 for his research work. He is a senior member of the IEEE and a member of the IEEE Computer Society.



Jie Chen received the MS and PhD degrees from the Harbin Institute of Technology, Harbin, China, in 2002 and 2007, respectively. Since September 2007, he has been a senior researcher in the Machine Vision Group at the University of Oulu, Finland. His research interests include pattern recognition, computer vision, machine learning, dynamic texture, human action recognition, and watermarking. He has authored more than 20 papers in journals and conferences and is a member of the IEEE.

Wen Gao received the PhD degree in electronics engineering from the University of Tokyo, Japan, in 1991. He is a professor of computer science at Peking University, China. Before joining Peking University, he was a professor of computer science at the Harbin Institute of Technology from 1991 to 1995, and a professor at the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively, including four books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He served or serves on the editorial board for several journals, such as the *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Autonomous Mental Development*, *EURASIP Journal of Image Communications*, and *Journal of Visual Communication and Image Representation*. He has chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations. He is a fellow of the IEEE.