



## Learning representations from multiple manifolds



Chan-Su Lee <sup>a,\*</sup>, Ahmed Elgammal <sup>b</sup>, Marwan Torki <sup>c</sup>

<sup>a</sup> Department of Electronic Engineering, Yeungnam University, Gyeongsan, Republic of Korea

<sup>b</sup> Department of Computer Science, Rutgers, The State University of New Jersey, New Brunswick, USA

<sup>c</sup> Department of Computer and System Engineering, Alexandria University, Alexandria, Egypt

### ARTICLE INFO

#### Article history:

Received 12 October 2014

Received in revised form

10 July 2015

Accepted 25 August 2015

Available online 5 September 2015

#### Keywords:

Manifold learning

Dimensionality reduction

Joint manifold representation

Correspondence

### ABSTRACT

The problem we address in this paper is how to learn joint representation from data lying on multiple manifolds. We are given multiple data sets, and there is an underlying common manifold among the different data sets. Each data set is considered to be an instance of this common manifold. The goal is to achieve an embedding of all the points on all the manifolds in a way that preserves the local structure of each manifold and that, at the same time, collapses all the different manifolds into one manifold in the embedding space while preserving the implicit correspondences between the points across different data sets. We propose a framework to learn embedding of such data, which can preserve the intra-manifolds' local geometric structure and the inter-manifolds' correspondence structure. The proposed solution works as extensions to current state-of-the-art spectral-embedding approaches to handling multiple manifolds.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

Dimensionality reduction techniques have proven useful in many computer vision problems. In particular, nonlinear dimensionality reduction techniques [1–7] can achieve embedding of data lying on a nonlinear manifold by changing the metric from the original space to the embedding space, based on the manifold's local geometric structure. Many of the introduced nonlinear dimensionality reduction techniques are instances of graph spectral-embedding [8]. Spectral-embedding approaches, in general, construct an affinity matrix between data points that reflects the local manifold structure, i.e., constructing a graph with edge weights reflecting the local geometry of the manifold. Embedding is then achieved through solving an eigenvalue problem on the affinity matrix. Examples of nonlinear dimensionality reduction techniques in this category include: isometric feature mapping (Isomap) [1], locally linear embedding (LLE) [2], Laplacian eigenmaps [3], and manifold charting [4], among others. Bengio et al. [9] and Ham et al. [10] showed that these approaches are all instances of kernel-based learning, in particular, kernel principle component analysis (KPCA).

All these nonlinear embedding frameworks were shown to be able to embed data lying on a nonlinear manifold into a low-dimensional Euclidean space for toy examples, as well as for real

images. Such approaches are able to embed image ensembles nonlinearly into low-dimensional spaces where various orthogonal perceptual aspects can be shown to correspond to certain directions or clusters in the embedding spaces. However, the application of such approaches is limited to embedding of a single manifold and, as we shall show, fails to embed data lying on multiple manifolds.

The problem we address in this paper is how to learn useful representation from data sets lying on multiple manifolds. We are given multiple data sets, and there is an underlying common manifold among the different data sets. Each data set is considered to be an instance of this manifold. For example, images of different objects with similar geometry are observed from different views, where the images of each object are one data set. The images of each object lie on the view manifold of that object, i.e., each data set is an instance of a view manifold of a different object. We can think of such manifolds as quasi-parallel in the space. Different objects' manifolds are distributed in the space. We can even think of such manifolds themselves as lying on a manifold (assume that we collapse each data set into a point). Another example comes from embedding motion manifolds for different people (e.g., consider data on different people walking). Each person's data set represents an instance of the "walking" manifold, while each person's walking manifold (as a whole) is lying on a different location in the input space. None of the existing manifold learning techniques can be used to learn such complex structures: both intra-manifold and inter-manifold structures.

\* Corresponding author.

E-mail address: [chansu@ynu.ac.kr](mailto:chansu@ynu.ac.kr) (C.-S. Lee).

**Our contribution:** We propose a framework for learning embedded representation from different data sets, each assumed to lie on a nonlinear manifold. The embedding achieves representation of the common underlying manifold shared between the different data sets. The problem of learning a common embedded representation from multiple data sets becomes trivial if we assume that there are given correspondences between the different data sets, i.e., the data sets are aligned. In this paper, we do not assume that such correspondences are given. The results we achieve are superior to existing state-of-the-art embedding approaches when applied to such a setting. The proposed solution works as extensions for the current state-of-the-art spectral-embedding approaches, such as Isomap [11], LLE [2], and Laplacian eigenmaps [3], to handle multiple manifolds.

The organization of the paper is as follows. [Section 2](#) discusses some motivating applications of the proposed approach. [Section 3](#) presents related works. [Section 4](#) presents the proposed joint manifold embedding with inter-manifold correspondence estimation. [Section 5](#) presents some examples of experimental results on different data sets.

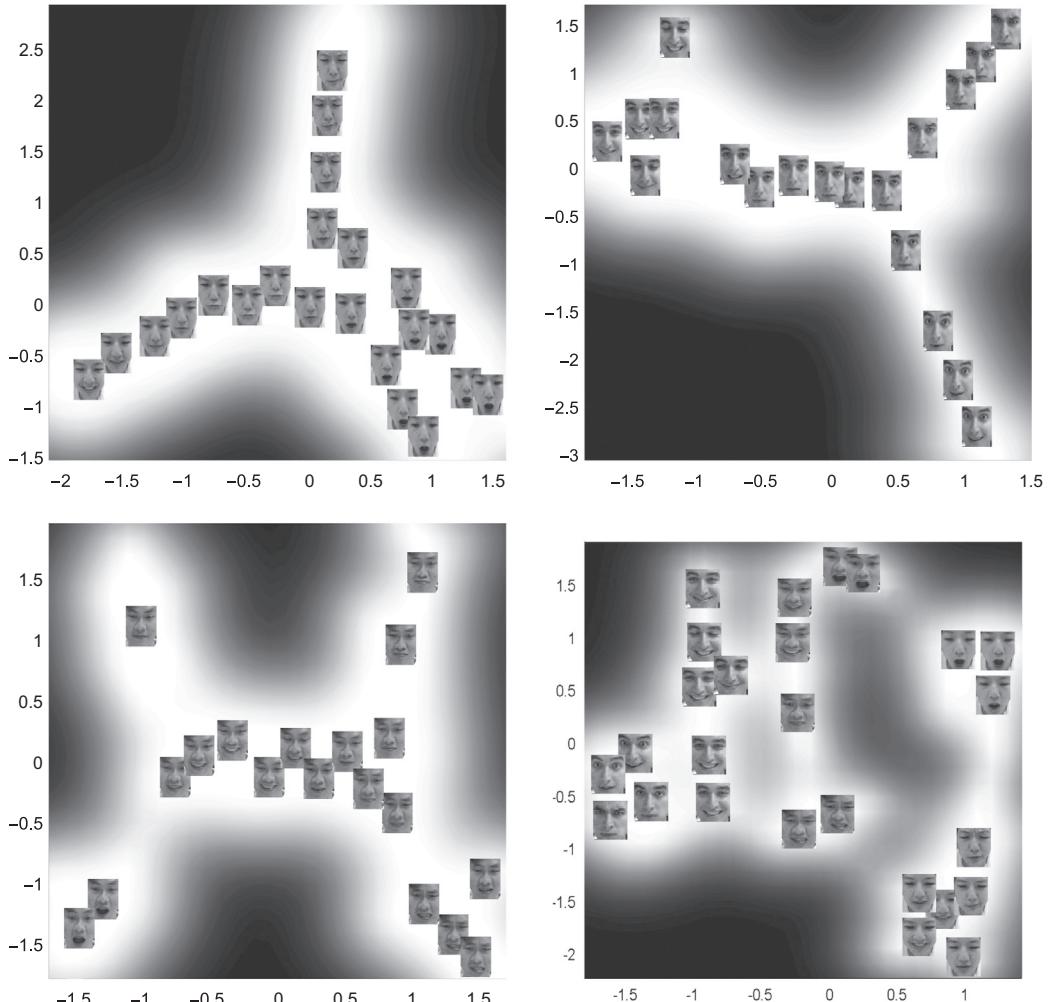
## 2. Motivation and problem definition

Suppose we are given  $K$  sets of data lying on  $K$  different manifolds, e.g., different people performing the same set of

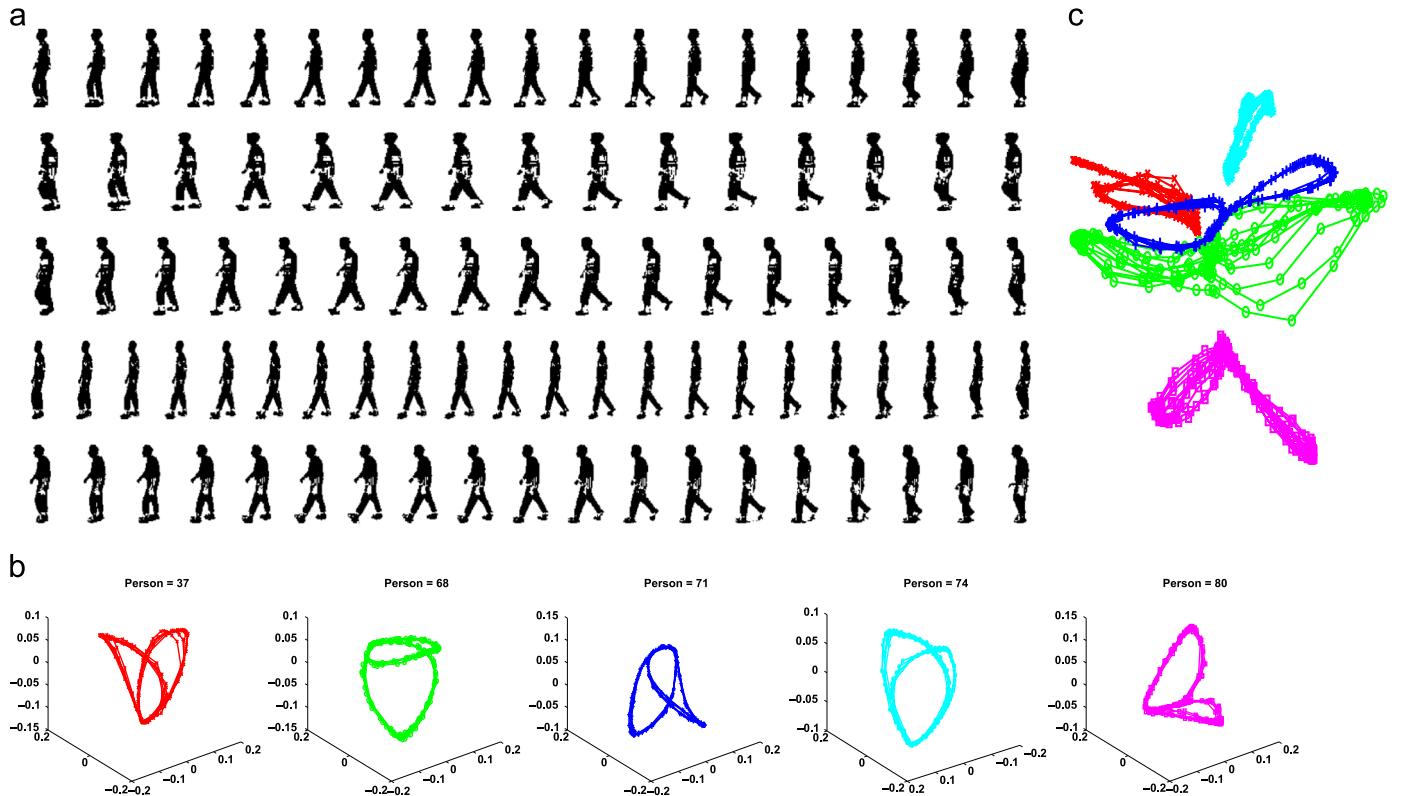
gestures or facial expressions. These data represent two different factors: body configuration variability, which typically lies on a low-dimensional manifold, and different people's variability, which might be higher in its dimensionality. One objective is to learn an embedding of the body-configuration manifold invariant to the person performing the motion. Learning such joint person-invariant body-configuration manifold embedding is essential for estimation of the intrinsic configuration, for providing dynamic models for tracking, and for recognition of gestures and activities.

Obviously, we can achieve embedding of each person's data set individually, which yields person-specific body configuration manifold embedding, as can be seen in [Fig. 1](#). Such embedding will be different from one person to another, and will not be useful in any general tracking or recognition task where the goal is to track and recognize the facial expression. On the other hand, if we put all the data together in one set and try to embed them using any nonlinear embedding technique, we will not be able to achieve meaningful embedding either, since the inter-manifold distance between data for different people will be much larger than the intra-manifold distance (within one specific person). So, we will end up with  $K$  separate clusters in the embedding space, as can be seen from the bottom-right embedding in [Fig. 1](#). In this example, a Gaussian process latent variable model (GPLVM) [5] is used to achieve the embedding.

Another example is shown in [Fig. 2](#), where we used shapes representing side-view walking sequences for multiple people



**Fig. 1.** Example embedding for facial expressions of three people: the first three plots show embedding for an individual person. The last plot shows embedding of three manifolds together, which is dominated by the inter-person manifold distance.



**Fig. 2.** Individual manifold embedding: (a) Input data. (b) Person-specific manifold embedding by Laplacian eigenmaps. (c) Laplacian eigenmap manifold embedding for all the data sets together.

from the CMU-Mobo Gait database [12]. An individual person's manifold embedding, as seen in Fig. 2(b), gives an excellent intuitive representation of the walking cycle as a one-dimensional manifold. On the other hand, if we put all the data together and embed them, a very messy and useless embedding is achieved, as seen in Fig. 2(c). The embedding is dominated by the inter-manifold distances, and does not reflect by any means the local intra-manifold structure of the manifold. Therefore, such embedding will not be useful in any general tracking or recognition task.

**Problem definition:** We are given multiple sets of data, each representing data lying on a nonlinear manifold where all these manifolds are conceptually similar, i.e., each set of data represents an instance of a common underlying manifold. The question is, how can we learn useful embedded representations of these data? How can we learn joint manifold embedding representing this common underlying manifold, regardless of the cross-manifold variability? All the manifolds are supposed to share the same geometric structure, and it is desired that embedding should reflect such a common geometric structure. In other words, we need to learn an “average” manifold.

We propose a framework to solve this fundamental problem. This problem has not been addressed before, and most of the research is focused on learning individual manifolds, assuming that all the data lie on a manifold. If correspondences among the manifolds are given, the problem becomes trivial. One solution in this case is to stack the data from the multiple sets together to form a joint space and embed the resulting high-dimensional (stacked) vectors. This is similar to the solution proposed by Ham et al. [13] for the problem of learning joint embedding across different spaces. We solve the problem within the embedding framework where, from the data geometry, we can achieve joint embedding. Fig. 3 shows that conventional manifold embedding can find nonlinear manifold characteristics from individual gait sequences but fails to find common characteristics from the

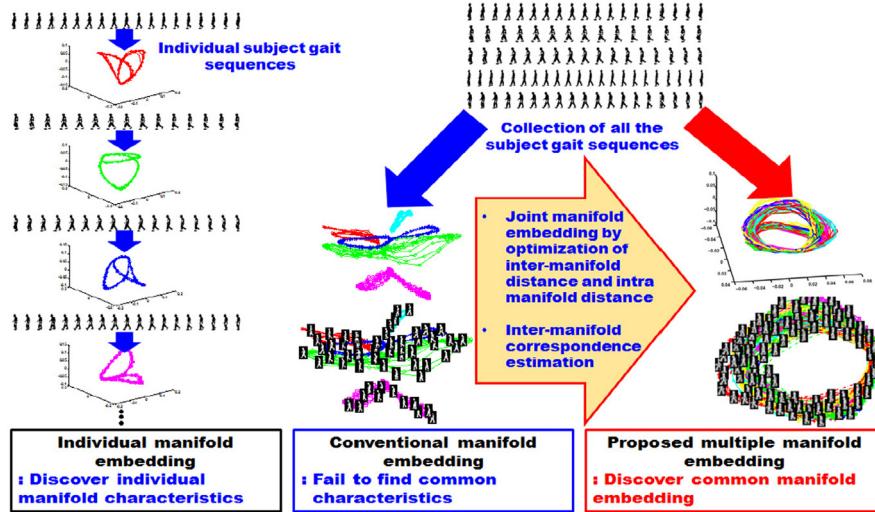
collection of gait sequences for multiple persons, whereas the proposed joint manifold embedding method can discover common manifold structures. Preliminary results of this work were presented in [14] and revised in this journal version with additional experimental results. The proposed solution works as extensions to current state-of-the-art spectral-embedding approaches, such as Isomap [11], LLE [2], and Laplacian eigenmaps [3], to handle multiple manifolds.

### 3. Related works

Traditional nonlinear manifold learning, such as Isomap, LLE, and Laplacian eigenmaps, fails to find a proper embedding space when there are disconnected multiple manifolds. Wu and Chan proposed an extension of Isomap embedding from separate computation of intra-class and inter-class geodesic distance for multiple manifold learning [15]. Souvenir and Pless presented another extension of Isomap embedding for multiple and intersection manifolds using manifold clustering [16].

Related to multiple manifold learning, we can categorize multiple cluster manifold learning and multiple-manifold learning; multiple cluster manifold learning is learning multiple manifolds that can be separated into different clusters, whereas multiple-manifold learning is learning overlapping manifolds in different structures. In the case of multiple cluster manifold learning, even though we cannot directly apply conventional manifold learning, we can solve the manifold learning problem by decomposition of all data into independent clusters and learn “inter-cluster” connections using conventional manifold learning. Further alignment or composition can be applied as post-processing to further align all manifold structures.

Manifold identification using multi-scale clustering algorithms was applied to multiple cluster manifold learning by Kushnir et al.



**Fig. 3.** Diagram to compare conventional manifold embedding for individual data sets and collection of data sets with the proposed joint manifold embedding.

[17]. Meng et al. [18] proposed a decomposition-composition (D-C) Isomap method for multiple cluster manifold learning. Neighborhood graph is used for decomposition. Wang et al. [19] showed that spectral clustering based on an affinity matrix of a local tangent angle and spatial neighborhood constraints can achieve clustering from multiple manifolds. Algorithm to resolving singularities was proposed for simultaneously achieve multiple manifold clustering and learning [20]. Recently, semi-supervised learning algorithms were also applied for manifold clustering with smoothness constraints [21], to discriminative metric learning from multiple manifold structures [22,23], and to predict the label of a data point according to its neighbors and to preserve intra-class local information [24].

When there are combinations of multiple factors, joint manifold learning or product manifold learning methods are developed for multiple manifold learning. In the case of product manifold learning, it assumes that pairwise combinations of data are available for data points on the manifold. For example, when we want to find joint manifold embedding for  $N$  objects and  $M$  views, we expect to have a pairwise combination of each object with  $M$  views. Therefore,  $N \times M$  samples are available for learning joint manifold embedding, as in the Columbia Object Image Library (COIL)-20 database.

Coupled visual and kinematic manifold learning based on view-invariant configuration manifolds and configuration-invariant view manifolds is an example of product manifold embedding using a marginalized embedding manifold [25]. Torus manifold embedding is also a product manifold based on embeddings of two conceptual circular manifolds and their product [26]. Cylindrical manifold embedding is used for topologically constrained low-dimensional manifold embedding from multiple locomotions at different speeds [27] or conceptual manifold embedding of joint views and hand-shape variations [28]. For multiview sequence data, intra- and inter-sequence neighbor constraints using representative views and their Laplacian eigenmaps are applied to find joint modeling of similar concepts [29].

Multilinear analysis frameworks and their kernel extensions are extended into Grassmann manifolds using principal angles as geodesic distance [30], which are parameterizations of average factors. To preserve characteristics of individual submanifolds, factor-dependent submanifolds are learned and local coordinates are aligned for a joint parameter space [31]. Product manifold embedding techniques are applied for action recognition [32] and temporal motion sequence analysis [33]. In human motion

analysis from video data, manifold learning with spatial and temporal constraints is applied for cyclic motion using a multiple kernel learning framework [34].

In the case of joint manifold learning, the data consist of multiple intersecting manifolds. Goldberg et al. presented a joint manifold learning algorithm using semi-supervised learning algorithms to separate different manifolds and supervised learning within each separated manifold [35]. Davenport et al. defined a joint manifold as a subset of the product manifold and applied it to sensor data fusion [36]. Groupwise manifold alignment was achieved via registration-based inter-dataset kernel for high-resolution dynamic magnetic resonance (MR) imaging [37]. A hierarchical mixture of density models was used for multiple manifold learning [38].

Alternatively, our proposed framework provides a way to preserve inter-manifold structure by forcing embedding coordinates of any corresponding points to be close together, in addition to preserving an intra-manifold structure by preserving the local manifold structure. An extension similar to our approach was proposed using an intra-manifold weight matrix and a soft correspondence matrix for the inter-manifold structure [39]. The difference is in the computation of the inter-manifold structure: we solve inter-manifold correspondences by finding a permuting orthogonal matrix constraint from inter-manifold geometry encoded by Gaussian kernels, whereas the other approach computes a soft correspondence matrix from the inter-manifold structure directly. In comparison, the experiment in Section 5.1.1 shows limitations when soft correspondence is estimated directly from the inter-manifold structure.

## 4. Joint embedding from multiple manifolds

### 4.1. Inter- and Intra-manifold structures

The input is  $K$  different data sets in a  $D$ -dimensional space, denoted by  $\mathbf{X}^k = \{\mathbf{x}_i^k \in \mathbb{R}^D, i = 1, \dots, N_k\}$ ,  $k = 1, \dots, K$ . The points on each data set are assumed to lie on a manifold, and there is a common structure between the different manifolds, i.e., each data set represents an instance of the manifold, which we intend to learn. Each data set might have a different number of points. Therefore, we denote by  $N_k$  the number of points in the data set  $k$ . Let  $N$  be the total number of points in all data sets, i.e.,  $N = \sum_{k=1}^K N_k$ . Throughout this paper, we will use superscripts to

indicate the manifold index (equivalently, the data set index) and subscripts to indicate the point index, i.e.,  $\mathbf{x}_i^k$  denotes point  $i$  on manifold  $k$ . The correspondences between the data sets are not known and, since the data sets are of different sizes, no one-to-one correspondences can be assumed.

In such data sets, there are two different geometric structures: the intra-manifold structure and the inter-manifold structure. We represent these geometric structures separately using two kernels: the intra-manifold kernel and the inter-manifold kernel, respectively, as detailed next.

#### 4.1.1. Intra-manifold structure

The intra-manifold structure is the geometric structure within a given manifold. All spectral-embedding approaches construct an affinity matrix  $\mathbf{W}$  between data points  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  that reflects the local manifold structure. Such a matrix represents the intra-manifold structure. The procedure for constructing this matrix differs, based on the approach. For example, Isomap [11] finds the shortest geodesic paths on the manifold. LLE [2] finds local linear weights to construct each point from its local neighbors. Laplacian eigenmaps [3] use a heat kernel given the nearest manifold neighbors. In all cases, the entries  $\mathbf{W}$  represent a special data-dependent kernel [9,10]  $K_D(\mathbf{x}_i, \mathbf{x}_j)$ . Given such a kernel matrix, embedding is achieved directly using the principle eigenvectors of  $\mathbf{K}_D$ . Here, we do not make any assumptions about how the intra-manifold structure is obtained.

Given  $K$  data sets, the objective is to preserve the intra-manifold structure or the local geometric structure within each manifold. Given the  $k$ th data set, we can construct an  $N_k \times N_k$  symmetric weight matrix  $\mathbf{W}^k$  representing its local geometric structure, as typically done in LLE, Isomap, etc. Therefore, given the  $K$  data sets, we have  $K$  weight matrices. This collection of matrices captures each manifold's local geometric structure. We denote this collection by the tensor  $\mathbf{W}$ , where

$$\mathbf{W}_{ij}^k = K_D(\mathbf{x}_i^k, \mathbf{x}_j^k).$$

#### 4.1.2. Inter-manifold structure

The inter-manifold structure is the geometric structure between the different data sets in the space. Given any two manifolds  $p$  and  $q$ , their pairwise inter-manifold structure is represented by an  $N_p \times N_q$  kernel matrix  $U^{pq}$ , such that

$$\mathbf{U}_{ij}^{pq} = G(\mathbf{x}_i^p, \mathbf{x}_j^q),$$

where  $G(\cdot, \cdot)$  is a global kernel between points  $\mathbf{x}_i^p$  and  $\mathbf{x}_j^q$  on manifolds  $p$  and  $q$ , respectively. This collection of matrices captures the pairwise manifold relations. We denote this collection by the tensor  $\mathbf{U}$ , where

$$\mathbf{U}_{ij}^{pq} = G(\mathbf{x}_i^p, \mathbf{x}_j^q)$$

Notice that the matrices  $\mathbf{U}^{pq}$  are not symmetric. However, there exists a hyper-symmetry structure in  $\mathbf{U}$  because  $\mathbf{U}_{ij}^{pq} = \mathbf{U}_{ji}^{qp}$ , i.e.,

$$\mathbf{U}^{pq} = \mathbf{U}^{qpT}.$$

#### 4.2. Objective function

Given the intra-manifold structures  $\mathbf{W}$  and the inter-manifold structure  $\mathbf{U}$ , the goal is to achieve an embedding of all the points on all the manifolds in a way that preserves the local structure of each manifold and, at the same time, collapses all the different manifolds into one manifold in the embedding space while preserving the implicit correspondences between the points across different data sets. Since we want to find common embedding among all the data sets, we do not need to preserve the

inter-manifold structure. The inter-manifold structure is needed to solve for soft correspondences between the data sets.

Formally, we seek an embedding for the data, i.e., we seek the coordinates of  $N$  points  $Y = \{\mathbf{y}_i^k \in \mathbb{R}^d, i = 1, \dots, N_k, k = 1, \dots, K\}$  in a  $d$ -dimensional embedding space where a point  $\mathbf{y}_i^k$  is the embedding of the data point  $\mathbf{x}_i^k$ . The desired embedding should satisfy three goals:

1. The embedding of any two points within the same manifold (data set) should preserve that manifold's local structure.
2. The embedding coordinates of any corresponding points on two different manifolds should be close together.
3. The embedding of any two points within the same manifold should preserve the local geometric structure of their corresponding points in all other manifolds.

The third goal is implicitly satisfied by the first two goals. That is, ensuring that corresponding points across different manifolds are embedded close to each other, while preserving the local structure of each manifold, guarantees the third goal.

If the correspondences between the different data sets are given, the problem becomes trivial, as will be shown. In our case, the correspondences are not given. Moreover, since the data sets are of different sizes, no optimal one-to-one correspondences are possible. Therefore, given the inter-manifold geometric structure, we aim to obtain an *inter-manifold soft correspondence structure* by solving for soft correspondences between the different data sets. The inter-manifold correspondence structure is denoted by  $C = \{C^{pq}\}$ , where a soft correspondence matrix between manifold  $p$  and manifold  $q$  is denoted by an  $N_p \times N_q$  matrix  $C^{pq}$ . There are different ways to find such soft correspondence matrices. Generally speaking, a high  $C_{ij}^{pq}$  value indicates a strong correspondence between point  $x_i^p$  and point  $x_j^q$ . Therefore,  $C^{pq} = C^{qpT}$ . In Section 4.3, we will show how such soft correspondence matrices can be obtained.

Given the above-stated goals, we reach the following objective function on the embedded points  $Y$ , which need to be minimized:

$$\phi(Y) = \sum_k \sum_{i,j} (\mathbf{y}_i^k - \mathbf{y}_j^k)^2 \mathbf{W}_{ij}^k + \sum_{p,q} \sum_{i,j} (\mathbf{y}_i^p - \mathbf{y}_j^q)^2 C_{ij}^{pq}. \quad (1)$$

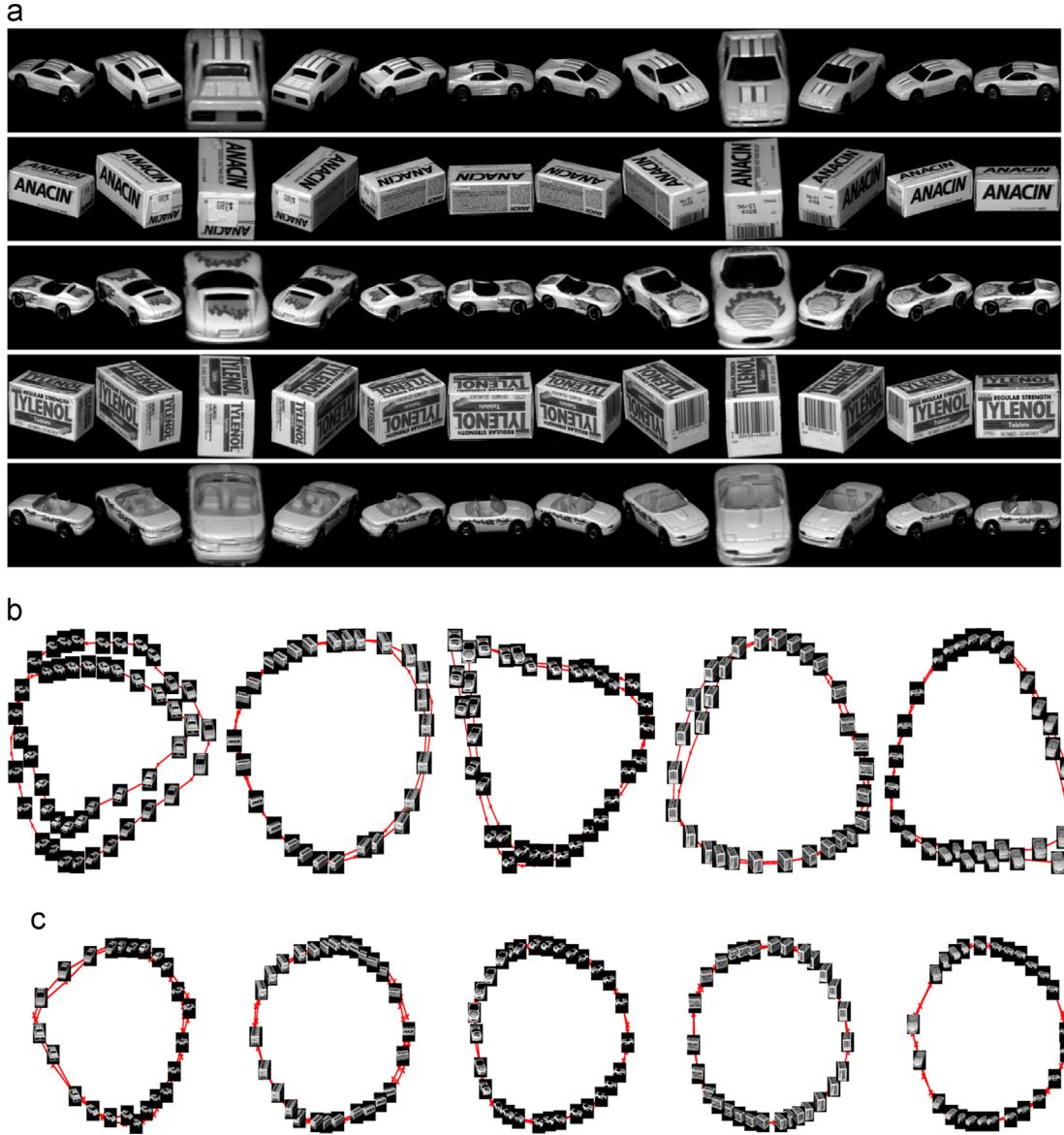
The objective function is intuitive. The first term of the objective function preserves the intra-manifolds' local geometry because it tries to keep the embedding  $(\mathbf{y}_i^k, \mathbf{y}_j^k)$  of any two points  $(\mathbf{x}_i^k, \mathbf{x}_j^k)$  on a given manifold close to each other based on their inter-manifold weight  $\mathbf{W}_{ij}^k$ . The second term of the objective function tries to bring close the embedded points  $(\mathbf{y}_i^p, \mathbf{y}_j^q)$  on manifolds  $p$  and  $q$  if their soft correspondence weight  $C_{ij}^{pq}$  is high. This objective function can be rewritten using one set of weights defined on the whole set of input points as

$$\phi(Y) = \sum_{p,q} \sum_{i,j} (\mathbf{y}_i^p - \mathbf{y}_j^q)^2 \mathbf{A}_{ij}^{pq}, \quad (2)$$

where the weight matrix  $\mathbf{A}$  is defined as

$$\mathbf{A}_{ij}^{pq} = \begin{cases} \mathbf{W}_{ij}^k & p = q = k \\ \mathbf{C}_{ij}^{pq} & p \neq q \end{cases} \quad (3)$$

This construction defines an  $N \times N$  weight matrix  $\mathbf{A}$  with  $K \times K$  blocks where the  $p - q$  block is of size  $N_p \times N_q$ . The  $p$ th diagonal block is the intra-manifold weight matrix  $\mathbf{W}^p$  for the  $p$ th manifold. The off-diagonal  $p - q$  block is the soft correspondence matrix  $\mathbf{C}^{pq}$ . The matrix  $\mathbf{A}$  is symmetric by definition because diagonal blocks are symmetric, and  $\mathbf{C}^{pq} = \mathbf{C}^{qpT}$ . The matrix  $A$  can be interpreted by the weights between nodes on a graph where all the input points



**Fig. 4.** COIL data set: examples of (a) input object appearance with view variations. (b) Object-specific view manifold embedding using LLE. (c) Object-specific view manifold embedding using Laplacian eigenmap.

are nodes on this graph. Nodes from a given data set are linked with weights representing their intra-manifold geometric structure, whereas nodes across different data sets are linked by weights representing their inter-manifold correspondence structure.

Given this construction, the objective function in Eq. (1) reduces to the problem of Laplacian embedding of a graph defined by the weight matrix  $\mathbf{A}$ . Therefore the objective function reduces to

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}^T \mathbf{D} \mathbf{Y} = 0} \text{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y}), \quad (4)$$

where  $\mathbf{L}$  is the Laplacian of the matrix  $\mathbf{A}$ , i.e.,  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is the diagonal matrix defined as  $D_{ii} = \sum_j A_{ij}$ . The  $N \times d$  matrix  $\mathbf{Y}$  is the stacking of the desired embedding coordinates such that,

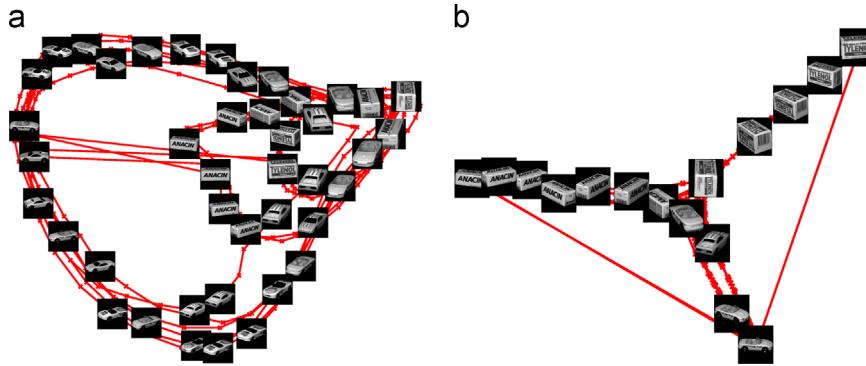
$$\mathbf{Y} = [y_1^1, y_2^1, \dots, y_{N_1}^1, y_1^2, y_2^2, \dots, y_{N_2}^2, \dots, y_1^K, y_2^K, \dots, y_{N_K}^K]^T.$$

Minimizing this objective function is a straight-forward generalized eigenvector problem:  $\mathbf{Ly} = \lambda \mathbf{Dy}$ . The optimal solution can be obtained by the bottom  $d$  nonzero eigenvectors. The required  $N$  embedding points  $\mathbf{Y}$  are stacked in the  $d$  vectors in such a way that

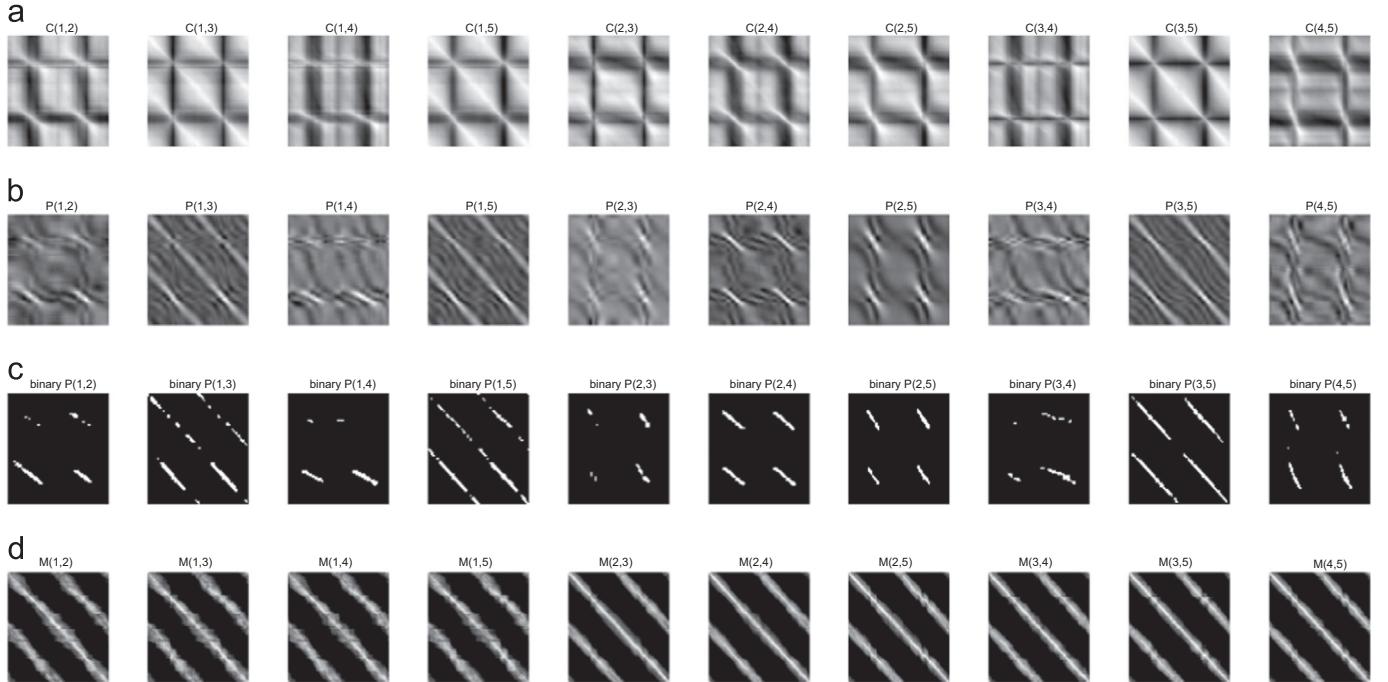
the embedding of the points of the first manifold will be the first  $N$  rows followed by the points of the second manifold, etc. (i.e., the obtained  $d$  eigenvectors are the stacking  $[y_1^1, y_2^1, \dots, y_{N_1}^1, y_1^2, y_2^2, \dots, y_{N_2}^2, \dots, y_1^K, y_2^K, \dots, y_{N_K}^K]^T$ ).

#### 4.3. Inter-manifold correspondences

Solving for correspondences between two data sets is a hard well-studied problem in computer vision. The difficulty is in finding a kernel weight matrix  $\mathbf{U}^{pq}$  between data sets  $\mathbf{X}^p$  and  $\mathbf{X}^q$  that is invariant to any geometric transformation on the two data sets. If a kernel invariant to geometric transformation is known, then the problem is a bipartite graph-matching problem, and therefore, efficient combinatorial algorithms, such as the Hungarian algorithm, can be used to obtain hard correspondences, i.e., a permutation matrix that brings the two sets into correspondence. Obviously, such geometric invariant kernels are not easy to achieve. Therefore, researchers have resorted to algorithms that can obtain soft correspondences between the data sets, i.e., relaxing the permutation matrix requirement. Ullman's minimal



**Fig. 5.** COIL data set: learning a joint view manifold from multiple objects. (a) LLE embedding for all the data. (b) Laplacian eigenmap embedding of all the data.



**Fig. 6.** COIL data set: learning a joint view manifold from multiple objects. (a) Pairwise correspondence matrices ( $C^{1,2}, C^{1,3}, C^{1,4}, C^{1,5}, C^{2,3}, C^{2,4}, C^{2,5}, C^{3,4}, C^{3,5}, C^{4,5}$ ). (b) Pairwise soft correspondence matrices ( $M^{1,2}, M^{1,3}, M^{1,4}, M^{1,5}, M^{2,3}, M^{2,4}, M^{2,5}, M^{3,4}, M^{3,5}, M^{4,5}$ ) [39].

mapping theory maximizes the inner product of a given matrix  $\mathbf{G}$  and its pairing matrix  $\mathbf{P}$  with an explicit constraint in linear programming [40]. Scott et al. proposed a pairing matrix  $\mathbf{P}$  for a given proximity matrix  $\mathbf{G}$  mutually orthogonal to maximize the inner product of  $\mathbf{PG}$  to model human perception in motion [41].

Given two data sets  $\mathbf{X}^p$  and  $\mathbf{X}^q$  with their inter-manifold geometric structure weight matrix  $\mathbf{U}^{pq}$ , maximum weight matching can be achieved by solving for a permutation matrix  $\mathbf{P}$  that permutes the rows of  $\mathbf{U}^{pq}$  in order to maximize its trace, i.e.,

$$\psi(\mathbf{P}) = \text{tr}(\mathbf{P}^T \mathbf{U}^{pq})$$

The permutation matrix constraint can be relaxed into an orthonormal matrix constraint on matrix  $\mathbf{P}$ . Therefore, the goal is to find an optimal orthonormal matrix  $\mathbf{P}^*$  such that

$$\mathbf{P}^* = \arg \max_{\text{s.t. } \mathbf{P}^T \mathbf{P} = \mathbf{I}} \text{tr}(\mathbf{P}^T \mathbf{U}^{pq}) \quad (5)$$

Scott et al. [41] showed that the optimal solution for Eq. (5) is

$$\mathbf{P}^* = \mathbf{UEV}^T$$

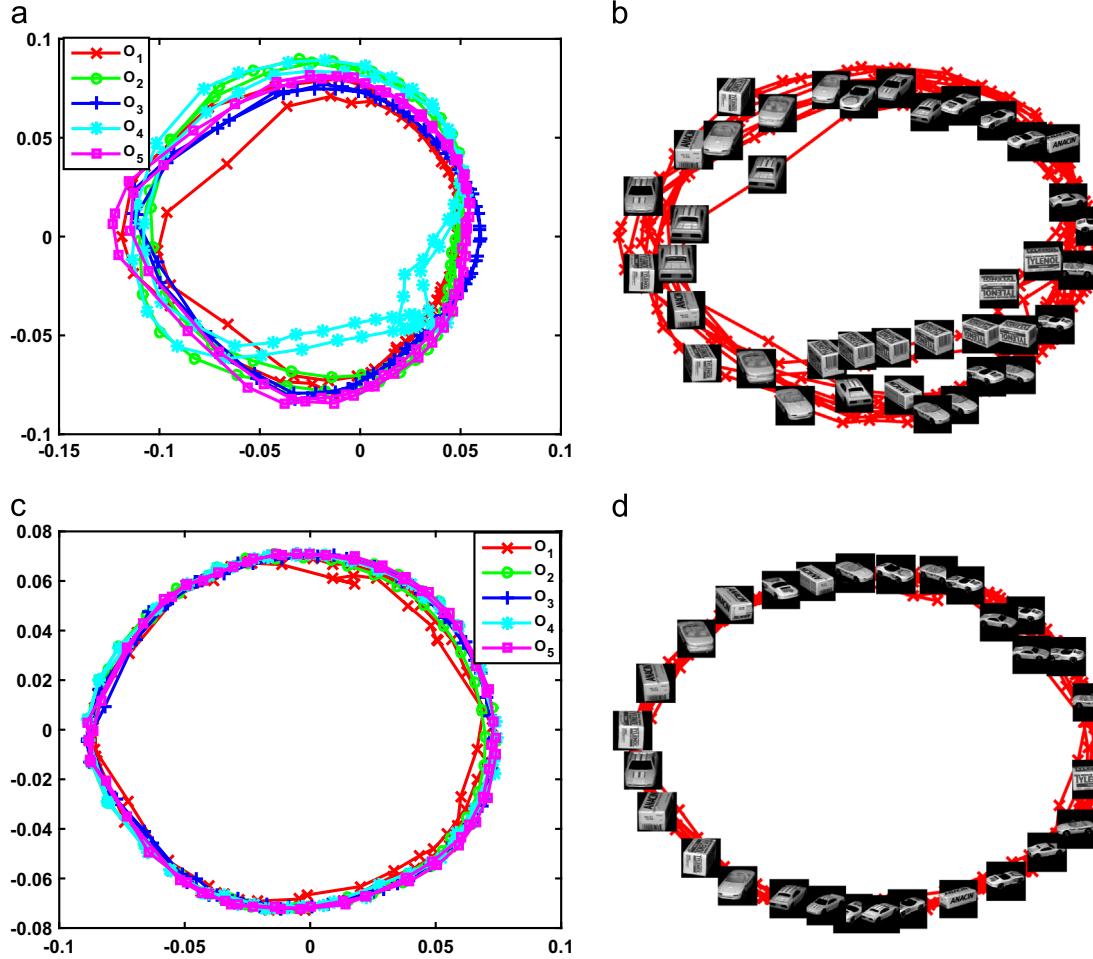
where the singular value decomposition (SVD) of  $\mathbf{U}^{pq} = \mathbf{USV}^T$ , and  $\mathbf{E}$  is obtained by replacing the singular values on the diagonal of  $\mathbf{S}$  by ones.

In our case, we have  $K_*(K - 1)/2$  inter-manifold weight matrices  $\mathbf{U}^{pq}$  and we need to obtain  $K_*(K - 1)/2$  correspondence matrices  $\mathbf{C}^{pq}$  that simultaneously maximize

$$\psi(C) = \sum_{p=1:K, q=p+1:K} \text{tr}(\mathbf{C}^{pq T} \mathbf{U}^{pq}) \quad (6)$$

Notice that the soft correspondences required for the embedding do not need to provide partitioning of the data, i.e., the correspondences do not need to be transitive. Solving for simultaneous hard correspondences is a weighted multipartite graph-matching problem, which is a much harder combinatorial problem. In our case, the objective function in Eq. (6) can be directly maximized by finding each of the pairwise correspondences  $\mathbf{C}^{pq}$  by solving Eq. (5).

We use a Gaussian kernel to encode the inter-manifold geometry, i.e.,  $\mathbf{U}_{ij}^{pq} = \exp(-\|\mathbf{x}_i^p - \mathbf{x}_j^q\|)/2\sigma^2$  where  $\sigma$  is a global scale that is estimated as a percentile of the overall data scale. Scott



**Fig. 7.** COIL data set: learning a joint view manifold from multiple objects. (a) Joint embedding of the view manifold obtained using the proposed algorithm. (b) Joint embedding with random sample images along the manifold. (c) Multiple manifold embedding of the view manifold obtained using [39]. (d) Multiple manifold embedding with random sample images along the manifold. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

et al. [41] showed that the Gaussian kernel fits the objective function for resolving the correspondences between image features related by an affine transformation, as long as no rotation component is involved. This conclusion is valid for data sets in higher dimensional spaces as well. In our case, since the data is high-dimensional, we expect the relation between the different manifolds to be close to an affine transformation with no much rotation involved. The experimental results we obtained confirm our speculation, because a Gaussian kernel was successful in bringing the different manifolds into correspondence in the embedding space. For partial intersection manifolds, the proposed estimation of soft correspondence may work well, with high correspondence in the intersection area. However, if the intersection regions are large, there may be confusion in the soft correspondence estimation.

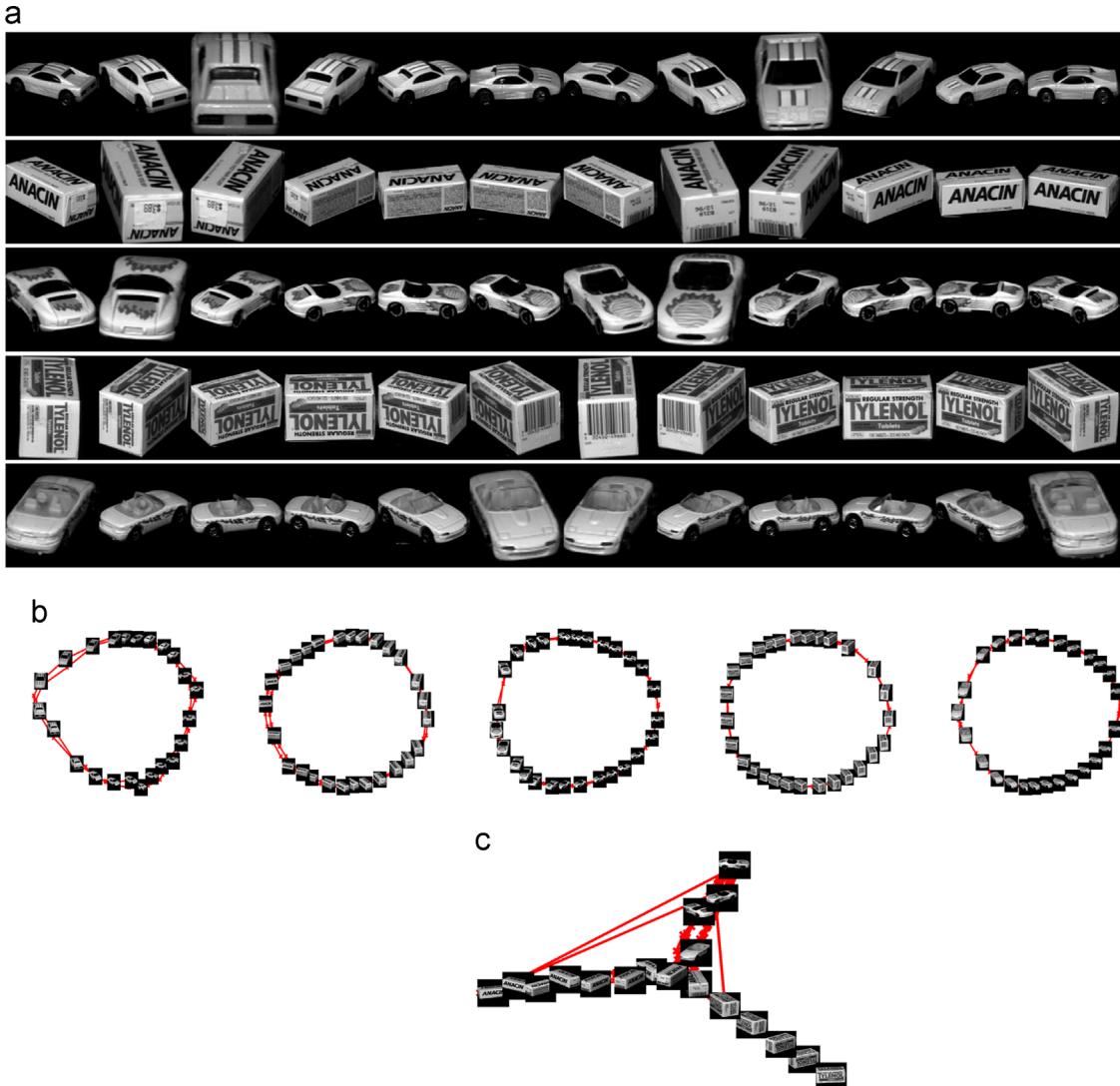
One obvious question arises: can the inter-manifold kernels  $\mathbf{U}^{pq}$  directly (or after scaling) be used in the objective function in Eq. (1) instead of  $\mathbf{U}^{pq}$ , since it also provides a measure of affinity between the points from the different data sets. The answer is no. The soft correspondences obtained by solving Eq. (5) incorporate the principle of exclusion because of the orthogonality constraints. However, to avoid direct estimation of the correspondence from high-dimensional sample data points, soft correspondence of inter-manifold geometry was estimated from the inter-manifold weight matrix in [39] as follows. The soft correspondence matrix

in each pair  $\mathbf{X}^c$  and  $\mathbf{X}^b$  is calculated by

$$M_{qj}^{cb} = \frac{\langle \mathbf{w}_q^c, \mathbf{w}_r^b \rangle}{\|\mathbf{w}_q^c\| \|\mathbf{w}_r^b\|}, \quad (7)$$

where  $\mathbf{w}_q^c \in \mathbb{R}^{1 \times n_c}$  and  $\mathbf{w}_r^b \in \mathbb{R}^{1 \times n_b}$  are row vectors of  $\mathbf{W}^c$  and  $\mathbf{W}^b$ . The assumption in this approach is that the inter-manifold similarity represents an intra-manifold correspondence, which can be easily violated, as shown in our experiment in Section 5.1.1. So it is necessary to estimate correspondences from a high-dimensional sample comparison and their transformation.

The computational time of the proposed approach can be analyzed by considering the computational steps compared with a Laplacian eigenmap. When the total sample number is  $N$  with  $K$  different data sets with  $N_k$ , a Laplacian eigenmap requires  $O(N^2)$  to construct a local neighborhood graph in the input space. In our proposed algorithm, we need to estimate  $K$  intra-manifold structures  $\mathbf{W}^k$  for each  $k$ . The computation time for an intra-manifold structure is  $O(KN_{k\max}^2)$ , where  $N_{k\max} = \max_k N_k$  is the maximum number of samples in each data set. If each data set has an equal number of sample data,  $N_k = N/K$  and  $N_{k\max} = N/K$ . In this case, the computational complexity of the intra-manifold structure is  $O(KN_{k\max}^2) = O(K(N/K)^2) = O(N^2/K)$ . Since each intra-manifold structure is a block-diagonal element of the total weight matrix, the computation time is  $K$  times less than the original total local neighborhood graph when each sample data set has an



**Fig. 8.** COIL data set with viewpoint shift: (a) Input object appearance with view variations starting from a shifted viewpoint. (b) Object-specific view manifold embedding using Laplacian eigenmap. (c) Laplacian eigenmap for all the collected sample data.

equal number of sample data. In addition, inter-manifold structure and inter-manifold correspondence have to be learned for  $K_*(K - 1)/2$  matrices with each  $N_p \times N_q$  size. The computational complexity for the inter-manifold structure can be computed with the computational complexity of  $O(K^2N_{k_{\max}}^2)$ . Inter-manifold correspondence can be estimated using SVD as in Eq. (5), which requires computational complexity  $O(K^2N_{k_{\max}}^3)$ . With an equal number of sample data, the computational complexity of inter-manifold complexity becomes  $O(K^2N_{k_{\max}}^3) = O(K^2(N/K)^3) = O(N^3/K)$ . After computation of  $N \times N$  weight matrix  $\mathbf{A}$  with  $K \times K$  blocks similar to the  $N \times N$  weight matrix  $\mathbf{W}$ , the generalized eigenvalue problems need to be solved with  $O(N^3)$  computational complexity in both Laplacian eigenmap and the proposed approach. Overall, in both cases, the computational complexity is the same  $O(N^3)$ . That is, the proposed approach does not require additional computational complexity compared with the Laplacian eigenmap.

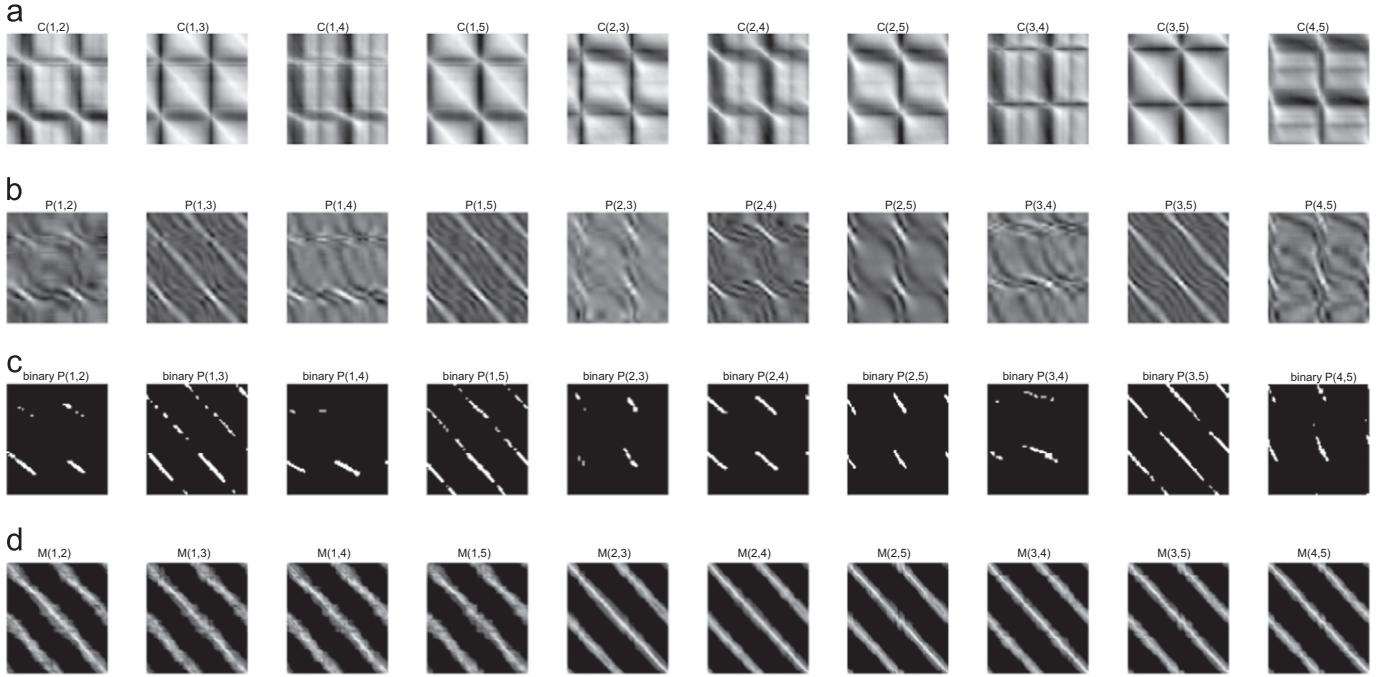
## 5. Experimental results

We ran experiments on different data sets. Here, we show experiments on four different data sets: object recognition data, facial expression data, human shape data, and human kinematic data.

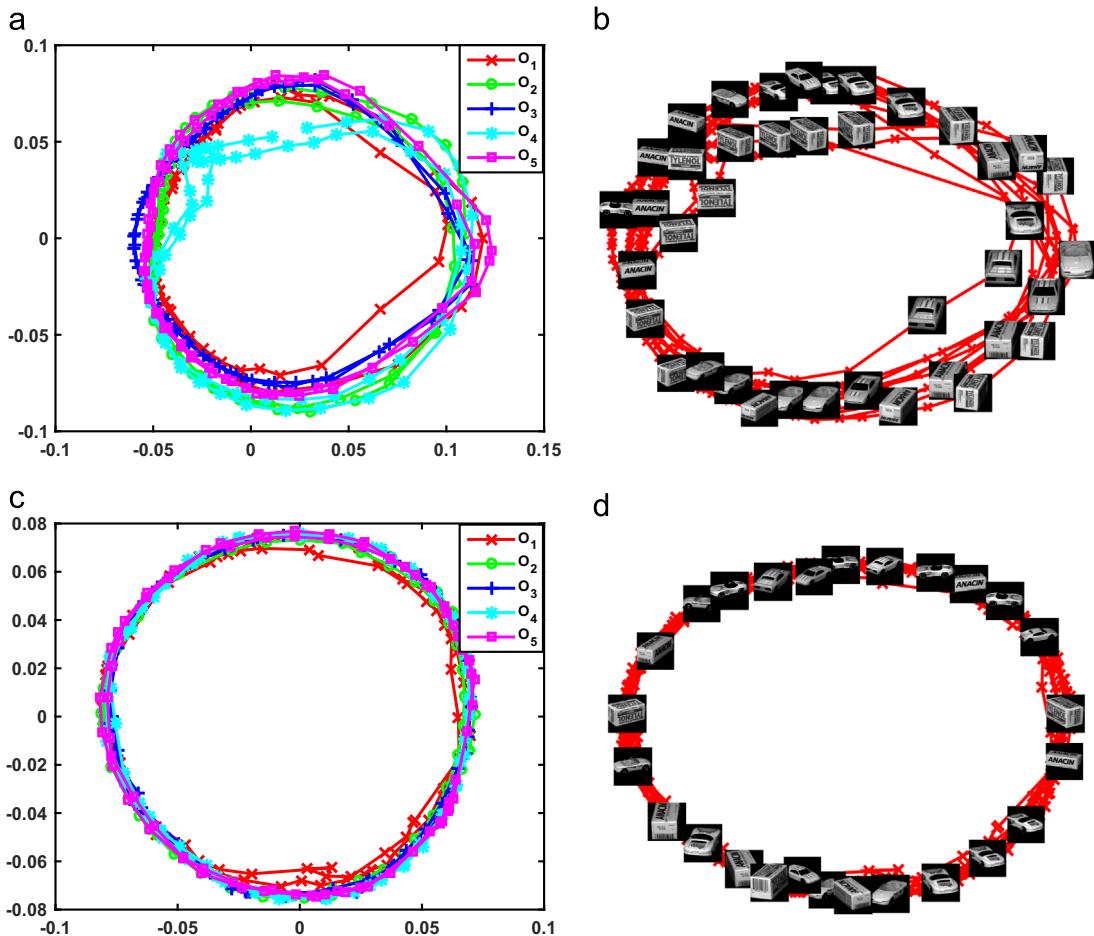
### 5.1. Example I: Learning a view manifold for different objects

In this experiment, the goal is to learn the visual view manifold from different objects' data. We used five objects with similar geometry from the COIL-20 data set [42] with different views for each object taken along a view circle. We used 72 views for each object, i.e., the data include 360 images. Each instance is a  $128 \times 128$  grayscale image. Although the different views are in correspondence in the data set, we did not use this correspondence information. Examples of the input data are shown in Fig. 4(a). Fig. 4(b) shows the individual manifold embedding for each object data obtained using LLE, and (c) using a Laplacian eigenmap. Putting all data together, traditional manifold embedding approaches fail to discover the structure of the data. This can be seen in Fig. 5(a) and (b) where LLE and Laplacian eigenmaps embedding results, respectively, are shown.

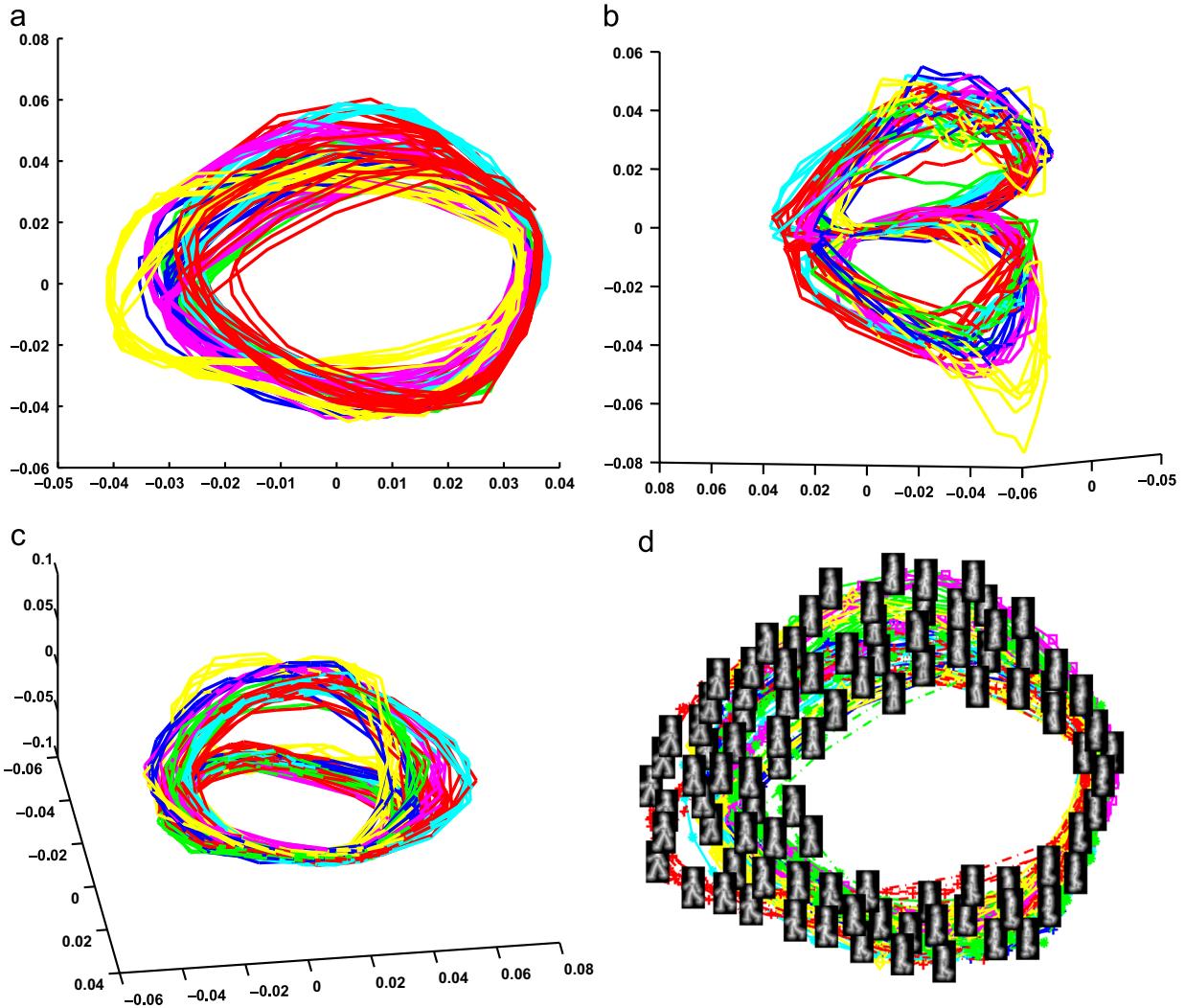
The proposed approach succeeds in learning a joint view manifold of all five objects. Fig. 6(a) shows pairwise inter-manifold structure among selected objects. There are 10 pairs from selected five objects. Fig. 6(b) shows estimated pairwise inter-manifold correspondence structure among selected objects. In the sample objects, the first object ( $O_1$ ), the third object ( $O_3$ ), and the fifth object ( $O_5$ ) are automotive with similar appearance. Therefore, the



**Fig. 9.** COIL data set with viewpoint shift: learning a joint view manifold from multiple objects without implicit alignment. (a) Pairwise correspondence matrices ( $C_{shift}^{1,2}, C_{shift}^{1,3}, C_{shift}^{1,4}, C_{shift}^{1,5}, C_{shift}^{2,3}, C_{shift}^{2,4}, C_{shift}^{2,5}, C_{shift}^{3,4}, C_{shift}^{3,5}, C_{shift}^{4,5}$ ). (b) Pairwise permutation matrices ( $P_{shift}^{1,2}, P_{shift}^{1,3}, P_{shift}^{1,4}, P_{shift}^{1,5}, P_{shift}^{2,3}, P_{shift}^{2,4}, P_{shift}^{2,5}, P_{shift}^{3,4}, P_{shift}^{3,5}, P_{shift}^{4,5}$ ). (c) Binalization of pairwise permutation matrices ( $binary\ P(1,2), binary\ P(1,3), binary\ P(1,4), binary\ P(1,5), binary\ P(2,3), binary\ P(2,4), binary\ P(2,5), binary\ P(3,4), binary\ P(3,5), binary\ P(4,5)$ ). (d) Pairwise soft correspondence matrices ( $M_{shift}^{1,2}, M_{shift}^{1,3}, M_{shift}^{1,4}, M_{shift}^{1,5}, M_{shift}^{2,3}, M_{shift}^{2,4}, M_{shift}^{2,5}, M_{shift}^{3,4}, M_{shift}^{3,5}, M_{shift}^{4,5}$ ).



**Fig. 10.** COIL data set with viewpoint shift: learning a joint view manifold from multiple objects with viewpoint shift. (a) Joint embedding of the view manifold obtained using the proposed algorithm. (b) Joint embedding with random sample images along the manifold. (c) Multiple manifold embedding of the view manifold obtained using the proposed algorithm. (d) Multiple manifold embedding with random sample images along the manifold.



**Fig. 11.** CMU gait example: embedding the gait manifold from seven subjects. (a), (b), and (c) are different views of the obtained three-dimensional embedding. (d) Visualization with sample images along the manifold. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

permutation matrix between the first object and the third object ( $P_{1,3}$ ), the first object and fifth object ( $P_{1,5}$ ), the third object and the fifth object ( $P_{3,5}$ ) show higher correspondence along diagonal axis than other pairs. Binarized matrices by thresholding the permutation matrices clearly show this trend as shown in Fig. 6(c). Pairwise soft correspondence matrices proposed in [39] are also shown in Fig. 6(d).

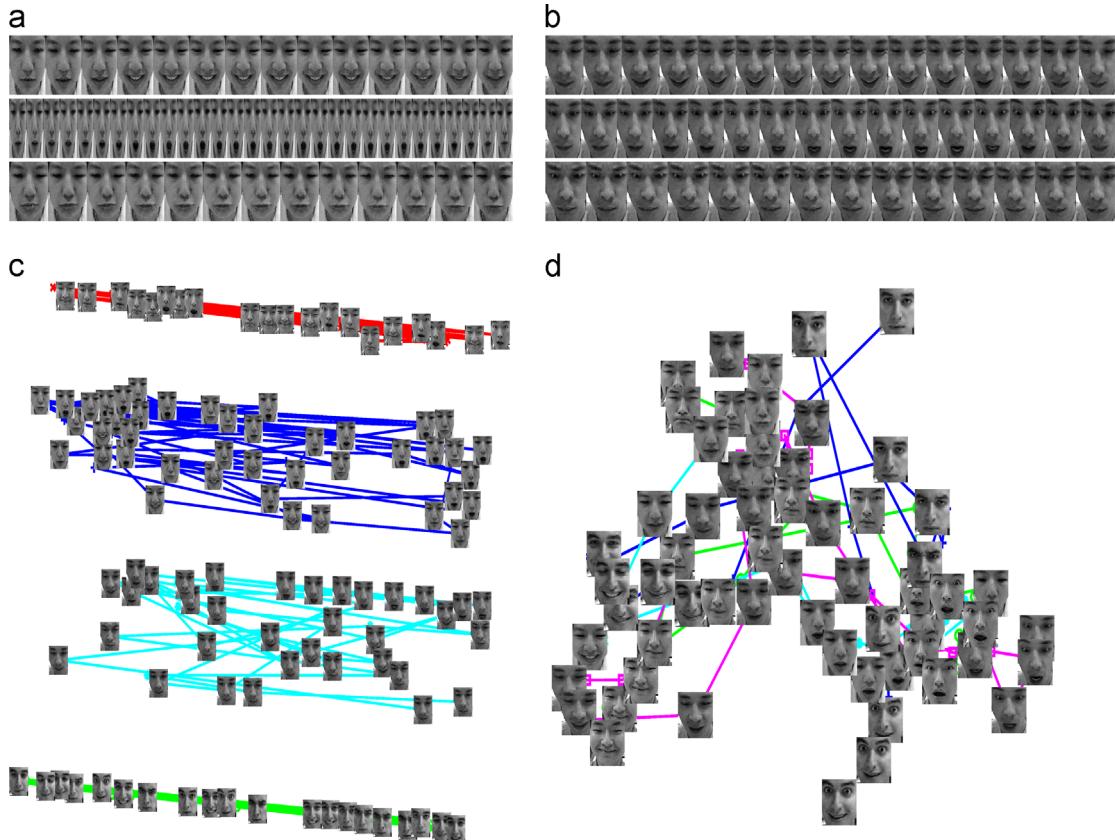
Fig. 7(a) shows the common view manifold obtained, where different line colors are used to represent different objects. Fig. 7(b) shows a visualization of the objects along the manifold using representative images, which were chosen at random from the five objects along the manifold. This clearly shows that all the objects are arranged in this representation according to the viewpoint and invariant of the object. Similar results can be achieved using the correspondence estimation from the embedding space when there are meaningful correspondences between the embedding points and actual observation data as shown in Fig. 7(c) and (d).

#### 5.1.1. Comparison with other multiple manifold learning: learning view manifolds for different objects with shifted view samples

In the previous experiment, the data set are implicitly aligned by view points in different objects. If viewpoint is shifted in a different object with a different offset, which remove implicit

alignment, can the presented framework or similar algorithm find correspondence in different objects? To answer this question, we collected each object data set after shifting view point as shown in Fig. 8(a). The first object has no shift, the second object view point was shifted by 20 degree (4 samples), the third object by 40 degree, and so on. The maximum offset value of the shifted view point was 80 degree compared with the original one. Fig. 8(b) shows the individual embedding manifolds for each object data obtained using Laplacian eigenmap, which are similar to the case of no shift in Fig. 4(c). Similar to the no shift case, putting all data together, the Laplacian eigenmap fails to discover the structure of the data that is shown in Fig. 8(c).

When we apply proposed approach to the shifted view data set, a joint view manifold of all five objects is learned similar to the no shift case. Fig. 9(a) shows pairwise correspondence matrices among selected objects. Fig. 9(b) shows estimated pairwise permutational matrices among selected objects. Viewpoint shift causes high off-diagonal correspondence. This tendency is clarified when we create binary matrices by thresholding the permutation matrices as shown in Fig. 8(c). Pairwise soft correspondence matrices proposed in [39] are also shown in Fig. 9(d), which does not show shift effect of view point in the correspondence estimation because the weight matrix does not change. We observe



**Fig. 12.** Embedding of joint facial expressions across different people. (a) and (b) Sample input data for two subjects. (c) Embedding of all the data with Laplacian eigenmaps. Notice the separation between the different people's manifolds. (d) Joint expression embedding obtained by the proposed approach.

similar individual manifold embedding with viewpoint shift in Fig. 8(b), and without shift of view point in Fig. 4(c), respectively.

Fig. 10(a) and (b) shows the common view manifold obtained using the proposed method. View point shifted object is now arranged well without shift effect by finding correct correspondence using estimated permutation matrix. However, the method in [39] fails to find correct alignment of manifold point according to view point. Some of the visualized nearby images show very different view point images in Fig. 7(d). When the algorithm estimate correspondence, the circular view embedding shows very small or no change in structure and fails to find shifting of the corresponding view points.

## 5.2. Example II: Learning a gait manifold for multiple subjects

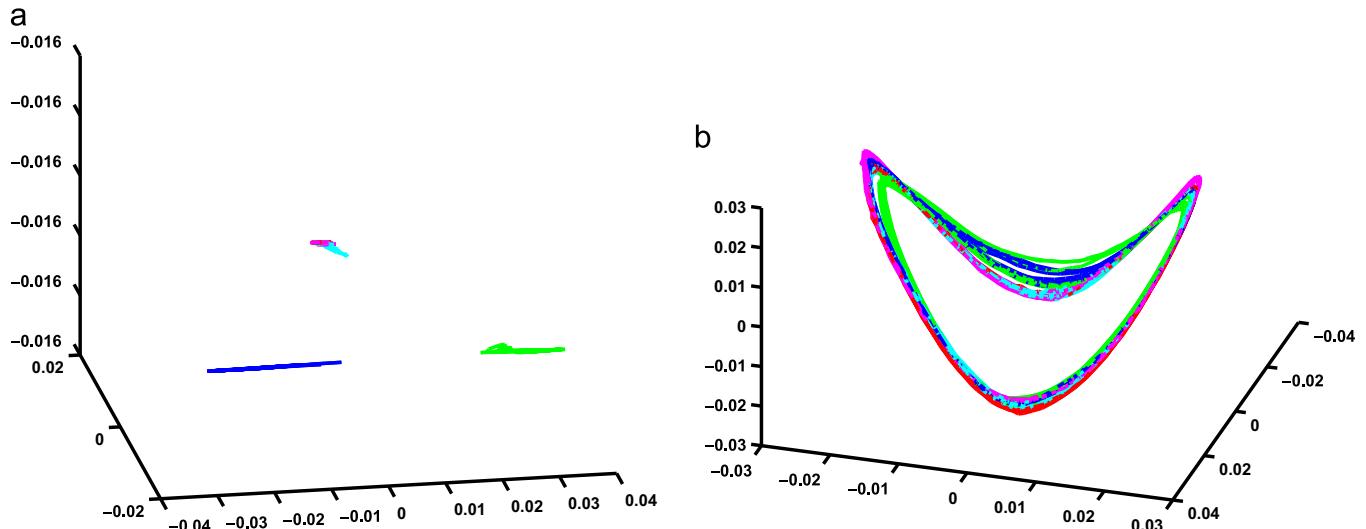
In this experiment, the goal is to learn embedding of the gait manifold from observations (silhouettes) of different people. We used data sets for different people's walking shapes from a side-view camera. The data are from the CMU-Mobo gait data set [12], and some examples of these data are shown in Fig. 2(a). For this experiment, we used data from seven people with six walking cycles each. Each input instance is an image sized  $100 \times 60$ . The number of frames in each data set varies from 189 to 234 (depending on the person's walking speed). This is an example of where the data sets are not necessarily in correspondence, with a different number of instances in each set. As was shown in Fig. 2(b), we can embed each person's data, which results in embedding that person's gait manifold. However, putting all the data together, LLE and other embedding techniques fail to obtain a useful embedded representation for the purpose of visualization or analysis, as shown in Fig. 2(c). The proposed approach can successfully learn joint embedding of the gait manifold across the

different subjects, invariant to the different people's shape variability, as shown in Fig. 11. The figure shows different viewpoints of the 3D embedding. The embedding shows a figure "8" embedding of the gait manifold. Different line colors mean embedding from different subjects in the data set. Fig. 11 (d) shows a visualization of the manifold with sample images chosen randomly from the input sets along the manifold. This is a two-dimensional projection of the three-dimensional embedding, and therefore, it is hard to visualize the images along the manifold. We can see similar body postures across different people along the manifold.

## 5.3. Example III: Learning a facial expression manifold for multiple subjects

In this experiment, the goal is to learn embedding of a joint facial expression manifold across different subjects. We used facial expression data from the CMU AMP facial expression database [43]. The data contains different people and different expressions. The input is four data sets. Each set contains a person performing three expressions (smiling, angry, surprised). Fig. 12(a) and (b) shows examples of the input images for two of the subjects. The number of images in each data set varies from 45 to 59 images. Each input instance is a  $64 \times 64$  grayscale images.

Fig. 12 (c) shows the embedding obtained using Laplacian eigenmaps with all data sets together. As can be seen in this figure, the embedding is dominated by the inter-manifold structure, and the embedding shows four separate clusters for the four subjects. LLE, regardless of the settings, also resulted in embeddings with four different clusters. Using the proposed approach, we can achieve common facial expression manifold embedding, invariant of the subject, and Fig. 12(d) shows the embedding obtained. We can clearly see that the three different expressions are located at



**Fig. 13.** Learning a kinematic locomotion manifold from different people and different running speeds. (a) Embedding achieved with Laplacian eigenmaps. (b) Joint embedding using the proposed approach. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

three different parts of the embedding space (the smiling expressions are in the lower left; the angry expressions are at the top; surprise is in the lower right; neutral is in the center of the embedding. Visualizing randomly chosen images on the manifold shows that the corresponding frames across the different subjects are close to each other).

#### 5.4. Example IV: Learning a kinematic locomotion manifold from different people

In this example, the goal is to learn a joint kinematic manifold for human locomotion from different subjects and from different running speeds. The input is 15 data sets from the EPFL motion capture data [44] for five subjects with three different speeds each (9, 10, and 11 km/h). Each data set has three running cycles with different numbers of input instances for a total of 3923 input frames. Each input instance is a 75-dimension motion-capture kinematic frame. As expected, although non-linear dimensionality reduction techniques can successfully achieve embedding of each data set alone, they fail, with different parameter settings, to achieve embedding of the 15 data sets together. An example is given in Fig. 13(a) for a Laplacian eigenmap embedding result. Five different colors (red, blue, green, cyan, and pink) are used to represent five different subjects even though the Laplacian eigenmap embedding does not show all the subject color because of overlapping in the point plot. The proposed approach successfully achieves embedding of the locomotion manifold, invariant of the different subjects and the different running speed variability. This is shown in Fig. 13(b).

## 6. Conclusion

In this paper, we introduced an approach for learning embedded representations from multiple manifolds. Given different sets of data lying on conceptually similar manifolds, we can achieve joint embedding of such common manifolds that preserves the intra-manifold's local structure. This is an important problem in data analysis where the goal is to learn an embedded representation of data from multiple sources (e.g., different people, objects, and spaces) regardless of the variability of the sources. We presented results from four real data sets, including object recognition data, visual gait data, facial expression data, and

kinematic locomotion data. In all cases we reached meaningful embedding of the underlying manifolds across different input data sets. The approach we introduced provides an important extension to state-of-the-art spectral-embedding techniques, such as LLE, Isomap, and Laplacian eigenmaps, for handling cases of multiple manifolds.

## Conflict of interest

None declared.

## Acknowledgments

This research was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1B4003830).

## References

- [1] J. Tenenbaum, Mapping a manifold of perceptual observations, in: Advances in Neural Information Processing, vol. 10, 1998, pp. 682–688.
- [2] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (5500) (2000) 2323–2326.
- [3] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* 15 (6) (2003) 1373–1396.
- [4] M. Brand, K. Huang, A unifying theorem for spectral embedding and clustering, in: Proceedings of the Ninth International Workshop on AI and Statistics, 2003.
- [5] N. Lawrence, Gaussian process latent variable models for visualization of high dimensional data, in: Proceedings of Advances in Neural Information Processing (NIPS), 2003.
- [6] K.W. Weinberger, L.K. Saul, Unsupervised learning of image manifolds by semidefinite programming, in: Proceedings of CVPR, vol. 2, 2004, pp. 988–995.
- [7] P. Mordohai, G. Medioni, Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting, in: Proceedings of International Joint Conference on Artificial Intelligence, 2005.
- [8] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 29 (1) (2007) 40–51.
- [9] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, M. Ouimet, Learning eigenfunctions links spectral embedding and kernel pca, *Neural Comput.* 16 (10) (2004) 2197–2219.
- [10] J. Ham, D.D. Lee, S. Mika, B. Schölkopf, A kernel view of the dimensionality reduction of manifolds, in: Proceedings of ICML, 2004, pp. 47–54.

- [11] J. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500) (2000) 2319–2323.
- [12] R. Gross, J. Shi, The cmu motion of body (mobo) database, Technical Report TR-01-18, Carnegie Mellon University, 2001.
- [13] J.H. Ham, D. Lee, L.K. Saul, Learning high dimensional correspondences from low dimensional manifolds, in: Proceedings of Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, ICML, 2003.
- [14] M. Torki, A.M. Elgammal, C.-S. Lee, Learning a joint manifold representation from multiple data sets, in: Proceedings of ICPR, 2010, pp. 1068–1071.
- [15] Y. Wu, K.K. Chan, An extended isomap algorithm for learning multi-class manifold, in: Proceedings of International Conference on Machine Learning and Cybernetics, vol. 6, 2004, pp. 3429–3433.
- [16] R. Souvenir, R. Pless, Manifold clustering, in: Proceedings of ICCV, 2005, pp. 648–653.
- [17] D. Kushnir, M. Galun, A. Brandt, Fast multiscale clustering and manifold identification, *Pattern Recognit.* 39 (10) (2006) 1876–1891.
- [18] D. Meng, Y. Leung, T. Fung, Z. Xu, Nonlinear dimensionality reduction of data lying on the multicluster manifold, *IEEE Trans. Syst. Man Cybern.-Part B: Cybern.* 38 (4) (2008) 1111–1122.
- [19] Y. Wang, Y. Jiang, Y. Wu, Z.-H. Zhou, Spectral clustering on multiple manifolds, *IEEE Trans. Neural Netw.* 22 (2) (2011) 1149–1161.
- [20] S. Martin, L. Szymanski, Singularity resolution for dimensionality reduction, in: Proceedings of International Conference on Image and Vision Computing, 2013, pp. 19–24.
- [21] Q. Wang, P.C. Yuen, G. Feng, Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions, *Pattern Recognit.* 46 (2013) 2576–2587.
- [22] M. Fan, X. Zhang, Z. Lin, Z. Zhang, H. Bao, Geodesic based semi-supervised multi-manifold feature extraction, in: 12th IEEE International Conference on Data Mining, 2012, pp. 852–857.
- [23] M. Fan, X. Zhang, Z. Lin, Z. Zhang, H. Bao, A regularized approach for geodesic-based semisupervised multi manifold learning, *IEEE Trans. Image Process.* 23 (5) (2014) 2133–2147.
- [24] M. Fu, B. Luo, M. Kong, Semi-supervised manifold learning based on 2-fold weights, *IET Comput. Vis.* 6(4), 348–354.
- [25] C.-S. Lee, A. Elgammal, Coupled visual and kinematic manifold models for tracking, *Int. J. Comput. Vis.* 87 (1) (2010) 118–139.
- [26] A.M. Elgammal, C.-S. Lee, Tracking people on a torus, *IEEE Trans. PAMI* 31 (3) (2009) 520–538.
- [27] R. Urtasun, D.J. Fleet, A. Geiger, J. Popović, T.J. Darrell, N.D. Lawrence, Topologically-constrained latent variable models, in: Proceedings of ICML, 2008, pp. 1080–1087.
- [28] C. Lee, S.Y. Chun, S.W. Park, Tracking hand rotation and various grasping gestures from an IR camera using extended cylindrical manifold embedding, *Comput. Vis. Image Underst.* 117 (12) (2013) 1711–1723.
- [29] M. Lewandowski, D. Mimitrios, S.A. Velastin, J.-C. Nebel, Structural Laplacian eigenmaps for modeling sets of multivariate sequences, *IEEE Trans. Cybern.* (2014) 936–949.
- [30] S.W. Park, M. Savvides, The multifactor extension of Grassmann manifolds for face recognition, in: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 2011, pp. 464–469.
- [31] S.W. Park, M. Savvides, Multifactor analysis based on factor-dependent geometry, in: Proceedings of CVPR, 2011, pp. 2817–2824.
- [32] Y.M. Lui, J.R. Beveridge, Action classification on product manifolds, in: Proceedings of CVPR, 2010, pp. 833–839.
- [33] M. Lewandowski, J.M. del Rincon, J.-C. Nebel, Temporal extension of Laplacian eigenmaps for unsupervised dimensionality reduction of time series, in: Proceedings of International Conference on Pattern Recognition, 2010, pdummyp, 161–164.
- [34] A. Álvarez Meza, J. Valencia-Aguirre, G. Daza-Santacoloma, C. Acosta-Medina, G. Castellanos-Domínguez, Video analysis based on multi-kernel representation with automatic parameter choice, *Neurocomputing* 100 (2013) 117–126.
- [35] A.B. Goldberg, X. Zhu, A. Singh, Z. Xu, R. Nowak, Multi-manifold semi-supervised learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, vol. 5, 2009, pp. 169–176.
- [36] M.A. Davenport, C. Hegde, M.F. Duarte, R.G. Baraniuk, High-dimensional data fusion via joint manifold learning, *IEEE Trans. Image Process.* 19 (10) (2010) 2580–2594.
- [37] C.F. Baumgartner, C. Kolbitsch, D.R. Balfour, P.K. Marsden, J.R. McClelland, D. Rueckert, A.P. King, High-resolution dynamic MR imaging of the thorax for respiratory motion correction of pet using groupwise manifold alignment, *Med. Image Anal.* 18 (2014) 939–952.
- [38] X. Wang, P. Tino, M.A. Fardal, Multiple manifolds learning framework based on hierarchical mixture density model, in: Principles of Data Mining and Knowledge Discovery, 2008, pp. 566–581.
- [39] J. Valencia-Aguirre, A. Álvarez Meza, G. Daza-Santacoloma, C. Acosta-Medina, G.C. Castellanos-Domínguez, Multiple manifold learning by nonlinear dimensionality reduction, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, LNCS 7042, 2011, pp. 206–213.
- [40] S. Ullman, dummy, *The Interpretation of Visual Motion*M.I.T. Press, Cambridge, 1976.
- [41] G.L. Scott, H.C. Longuet-Higgins, An algorithm for associating the features of two images, *Proc.: Biol. Sci.* 244 (1991) 21–26.
- [42] S.A. Nene, S.K. Nayar, H. Murase, Columbia object image library (coil-20), Technical Report CUCS-005-96, Columbia University, 1996.
- [43] X. Liu, T. Chen, B.V. Kumar, Face authentication for multiple subjects using eigenflow, *Pattern Recognit.* 36 (2003) 313–328.
- [44] R. Urtasun, P. Glardon, R. Boulic, D. Thalmann, P. Fua, Style-based motion synthesis, *Comput. Graph. Forum* 23 (4) (2004) 799–812.

**Chan-Su Lee** received the B.S. degree in Electronics Engineering from Yonsei University in 1995 and the M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1997, and the Ph.D. degree in Computer Science from Rutgers, The State University of New Jersey, in May 2007. From 1997 to 2001, he was a member research engineer at the Electronics and Telecommunications Research Institute (ETRI). He is currently an associate professor in the Department of Electronic Engineering at Yeungnam University, Korea. His research interests include computer vision, pattern recognition, machine learning, biometrics, gesture and facial expression analysis, smart lighting control and brain-computer interface. He has been a visiting professor at the University of Maryland, College Park, from 2013 to 2014. He is currently the director of Automotive Lighting LED-IT Convergence Education (ALICE) funded by Ministry of Trade, Industry & Energy.

**Ahmed Elgammal** received his B.Sc. and M.Sc. degrees in Computer Science and Automatic Control from the University of Alexandria, Egypt in 1993 and 1996, respectively. He received another M.Sc. and his Ph.D. degrees in Computer Science from the University of Maryland, College Park in 2000 and 2002, respectively. Currently, Dr. Ahmed Elgammal is an associate professor at the Department of Computer Science, Rutgers, the State University of New Jersey since Fall 2002. Dr. Elgammal is also a member of the Center for Computational Biomedicine Imaging and Modeling (CBIM) and the Center for Advanced Information Processing (CAIP) at Rutgers. His primary research interest is in computer vision and machine learning. His research focus includes human activity recognition, human motion analysis, tracking, human identification, and statistical methods for computer vision. He develops robust real-time algorithms to solve computer vision problems in areas such as visual surveillance, visual human-computer interaction, virtual reality, and multimedia applications.

**Marwan Torki** is an Assistant Professor in the Computer and Systems Engineering Department at Alexandria University, Egypt. He is also lecturing in VT-MENA (Virginia Tech-Middle East and North Africa) and PUA (Pharos University at Alexandria). He received his B.Sc. and M.Sc. in Computer Science from Alexandria University in 2003 and 2006, respectively and his Ph.D. in Computer Science from Rutgers University NJ, USA in 2011. He is advising several undergraduate/graduate students in different research and development projects focusing mainly on computer vision and machine learning tasks. Dr. Marwan Torki research wide interests are computer vision and machine learning. His research focus includes understanding, feature matching, activity recognition, manifold learning, object recognition, detection and Arabic NLP.