Learning nonlinear image manifolds by global alignment of local linear models

Jakob Verbeek

Learning Nonlinear Image Manifolds by Global Alignment of Local Linear Models

Jakob Verbeek

Abstract—Appearance-based methods, based on statistical models of the pixel values in an image (region) rather than geometrical object models, are increasingly popular in computer vision. In many applications, the number of degrees of freedom (DOF) in the image generating process is much lower than the number of pixels in the image. If there is a smooth function that maps the DOF to the pixel values, then the images are confined to a low-dimensional manifold embedded in the image space. We propose a method based on probabilistic mixtures of factor analyzers to 1) model the density of images sampled from such manifolds and 2) recover global parameterizations of the manifold. A globally nonlinear probabilistic two-way mapping between coordinates on the manifold and images is obtained by combining several, locally valid, linear mappings. We propose a parameter estimation scheme that improves upon an existing scheme and experimentally compare the presented approach to self-organizing maps, generative topographic mapping, and mixtures of factor analyzers. In addition, we show that the approach also applies to finding mappings between different embeddings of the same manifold.

Index Terms—Feature extraction or construction, machine learning, statistical image representation.

1 Introduction

Over the last two decades, appearance-based methods have become increasingly popular to solve computer vision problems such as object recognition and pose estimation [1]. These methods are based on statistical models of the pixel gray or color values rather than on geometrical models of the depicted objects. One can think of the images as vectors in a high-dimensional image space which is spanned by the values of the individual pixels. Note that the dimensionality of the image space is typically large, e.g., when modeling small 64×64 pixel images, the image space has 4,096 dimensions. It turns out that in many applications, the images of interest are confined to a low-dimensional manifold embedded in the high-dimensional image space. Our interest in this paper is to model image collections sampled from such manifolds. We build upon a recent method [2] to 1) model the density of images sampled from such manifolds and 2) recover global parameterizations of these manifolds in an unsupervised manner. A globally nonlinear probabilistic mapping between coordinates on the manifold to images—and vice versa—is obtained by combining several, locally valid, linear mappings.

In general, reliable estimation of the parameters of densities that implement an appearance-based model requires many images due to the high-dimensionality of the image space. To alleviate this problem, often, the simplifying assumption is made that, although the images are high-dimensional, the set of images considered in a given application is confined to a low-dimensional subspace of the high-dimensional image space. This idea was introduced in [3], where principal component analysis (PCA) [4] was used to identify, for a given set of images, a linear subspace which contains most of the variance in the images. The low-dimensional representation obtained by projecting an image on this subspace then gives an informative characterization of the actual image. Later, others proposed to use the projections on a low-dimensional linear subspace for tasks such as recognition and pose estimation [5], [6]. The idea that the images are confined to a low, say d, dimensional subspace of the image space (say of dimension D) is supported by the following reasoning. If there are only $d \ll D$ degrees of freedom (DOF) in the system that generates the images, and small changes in the DOF lead to small changes in the images, then the images will be confined to a d-dimensional manifold smoothly embedded in the D-dimensional image space.[1]

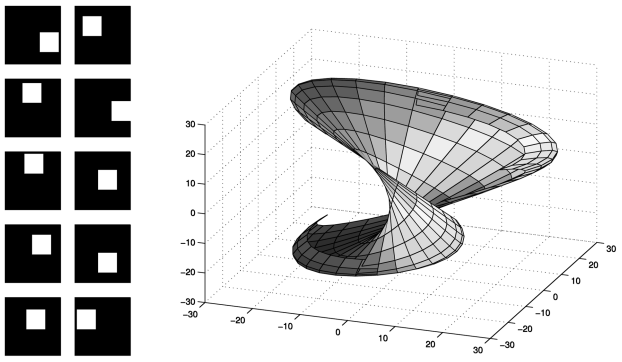
In many applications, images have indeed been found to be confined to a low-dimensional manifold. However, the manifold is often nonlinearly embedded in the image space. Fig. 1 gives a simple illustration. A set of 400 images of 29×29 pixels containing a white square of 10×10 pixels against a black background is considered. Here, there are two DOF: corresponding to the horizontal and vertical position of the white square. Fig. 1a depicts the projections of the images on the first three principal components (note: this projection is linear). Images where the position of the square differs only one pixel are connected to reveal the manifold structure. Clearly, the manifold spanned by the DOF is nonlinearly embedded in the image space.

This paper is concerned with modeling the DOF by recovering, or "learning," the underlying manifold structure given just a set of high-dimensional points (images) sampled from this manifold. In specific, we want to find a function that maps points in the image space to the corresponding low-dimensional coordinates on the manifold and vice versa. In this manner, we obtain a compact representation of the images in terms of the DOF rather than the, somewhat

• *The author is with GRAVIR-INRIA, 655 avenue de l'Europe, 38330 Montbonnot, France. E-mail: verbeek@inrialpes.fr.*

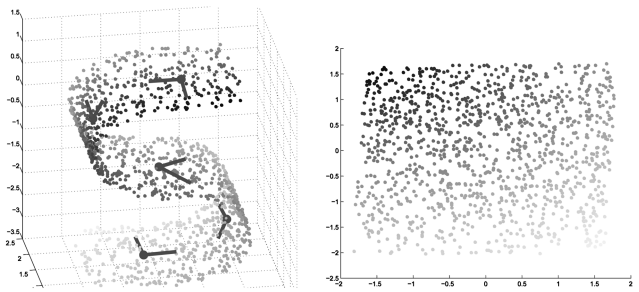
Manuscript received 24 Feb. 2005; revised 25 Nov. 2005; accepted 19 Dec. 2005; published online 13 June 2006.
Recommended for acceptance by A. Srivastava.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0102-0205.

1. With the DOF in the system that generates the images, we mean both those in the imaged object and those in the camera.



(a) (b)

Fig. 1. (a) Ten randomly selected images of the shifted square. (b) Images projected on the first three principal components, projections of images of similar shifts are connected.



(a) (b)

Fig. 2. (a) Data in $\mathbb{R}^3$ with local two-dimensional subspaces indicated by the axes. (b) The desired global two-dimensional data representation unfolds the nonlinear manifold.

arbitrary and redundant, coordinates of the image space. The learning task is "unsupervised" in the sense that, for none of the images, the coordinates on the manifold are known in advance.

A wide range of techniques has been proposed for unsupervised learning of nonlinear manifolds, such as autoencoder neural networks [7], principal curves and surfaces [8], [9], [10], self-organizing maps [11], [12], and generative topographic mapping [13]. For more extensive overviews, see [14], [15], [16]. All these methods suffer from one or more of the following drawbacks: 1) parameter estimation can return, suboptimal solution since it is based on minimizing an error function with local minima that are not global minima and 2) there is no notion of the uncertainty in the estimated low-dimensional coordinates of the images. Recently, various techniques, such as Isomap [17], Locally Linear Embedding (LLE) [18], Kernel PCA [19], charting [20], Locality Preserving Projections [21], [22], Laplacian Eigenmaps [23], and a semidefinite programming approach [24], have been proposed that formalize manifold learning as minimizing a convex error function that encodes how well certain interpoint relationships are preserved. Due to the convexity of the error function, there is no risk that minimization yields poor local minima rather than the global minimum of the error function, so drawback 1) is resolved for these methods. However, these methods do suffer from drawback 2), mentioned above. In addition, these methods do not provide a parametric function that maps between the image space and the low-dimensional manifold. Nonparametric methods can be used to map between the spaces, but, in principle, they require storage and access to all training data [25], which is costly for large high-dimensional data sets. If the goal is to find the manifold structure in a given training set only, e.g., for visualization, then this is not a problem. However, if we want to generalize the recovered manifold structure to new data (i.e., represent new images in terms of the DOF), then this is problematic because of storage and computational requirements.

In this paper, in contrast to these nonparametric methods, we consider a probabilistic model that combines several, locally valid, linear mappings to define a globally nonlinear mapping between the image space and the low-dimensional manifold. Once the parameters have been estimated from a set of images, a nonlinear mapping is obtained which can be used to map previously unseen images $\mathbf{x}$ to their low-dimensional coordinates on the manifold $\mathbf{z}$ and vice versa, without the need to store the original training data. In fact, the model yields a conditional density on $\mathbf{z}$ given $\mathbf{x}$ and vice versa and, thus, also gives a notion of uncertainty in the mapping. We note here that the required amount of training data to learn the model necessarily increases with the dimensionality of the manifold, for reasons of numerical stability.[2]

Intuitively, the approach taken in this paper can be understood as a combination of clustering and PCA. The idea is illustrated in Fig. 2; although, globally, the data is spread out in all three dimensions, locally, the data is concentrated around a two-dimensional linear subspace. Thus, the data can be clustered in such a manner that the data within each cluster can be accurately reconstructed from a two-dimensional PCA projection. Several authors, e.g., [26], [27], [28], have reported that such a combination of clustering and PCA allows for significantly better reconstructions of images as compared to a reconstruction using one single linear PCA subspace. Others [29] successfully used such a "mixture of PCAs" (MPCA) to classify images of hand written digits; an MPCA model was learned on images of each digit $i = 0, \ldots, 9$, yielding density models p_i $(i = 0, \ldots, 9)$. The learned models were used to classify new images $\mathbf{x}$ of handwritten digits by classifying them as digit $i = \arg\max_j p_j(\mathbf{x})$.

An important observation is that in all these works [26], [27], [28], [29], each cluster provides a *separate* low-dimensional coordinate system that is only used for data in that cluster. Therefore, it is not possible to directly interpolate between images that are assigned to different clusters since a global low-dimensional structure is lacking. Finding a global low-dimensional representation is also useful as a preprocessing stage for supervised learning problems since the low-dimensionality safeguards against the curse of dimensionality [30]. Thus, once a global low-dimensional representation is found based on many unsupervised examples, it is possible to learn functions on the manifold using relatively few supervised examples.

2. However, to our knowledge, it is unclear how the amount of data should depend on the manifold-dimensionality. In our experiments, with manifold-dimensionalities up to three, we did not encounter numerical instabilities.

In the next section, we describe probabilistic mixtures of factor analyzers and how they can be used to obtain a global low-dimensional representation. We present a parameter estimation technique, which improves upon earlier work by Roweis et al. [2]. Then, in Section 3, we show how this approach applies to correspondence learning, a recently introduced learning problem [31]. The learning data consists of two sets of high-dimensional points related by correspondences, i.e., some points in one set are known to have the same low-dimensional coordinate as a point in the other set. For example, each set could contain images of a different object as viewed from different directions. The correspondences are pairs of images where the two objects are viewed from the same direction. Nonlinear dimension reduction for the two sets of points simultaneously allows us to predict the correspondences for other points. In Section 4, we present the experimental results to evaluate the presented methods. Finally, in Section 5, we present our conclusions.

2 Modeling Data Sampled from Manifolds

In this section, we describe a probabilistic model for data sampled from nonlinear manifolds. In Section 2.1, we describe linear subspace densities which can be combined in a mixture density to model nonlinear manifolds. Then, in Section 2.2, we describe a framework to simultaneously estimate the parameters of such mixtures and to obtain a global parameterization of the manifold. In Section 2.3, we present our improved parameter estimation procedure. Since the optimization procedure is only guaranteed to find local optima of the objective function, careful initialization of the parameters is required. In Section 2.4, we consider parameter initialization methods.

2.1 Mixtures of Linear Models

We assume that the high-dimensional data vectors $\mathbf{x}_n \in \mathbb{R}^D$ $(n = 1, \ldots, N)$ are sampled independently and identically from a smooth nonlinear manifold and are potentially corrupted by additive uncorrelated Gaussian noise. Thus, the data is distributed on, or near, a low-dimensional manifold embedded in a high-dimensional space. If the manifold is sufficiently smooth, i.e., locally, it is linearly embedded in the high-dimensional space, then we can model the data density with a mixture of densities that concentrate their mass in a linear subspace. To this end, we can use mixtures of factor analyzers (MFA).

An MFA density over data vectors $\mathbf{x} \in \mathbb{R}^D$ can be specified by introducing, for each observed data vector two, "hidden" (or "unobserved") variables: c and $\mathbf{z}$. The first variable, $c \in \{1, \ldots, C\}$, is an index over the C components of the mixture. The second variable $\mathbf{z} \in \mathbb{R}^d$ is a coordinate in the d-dimensional subspace associated with the mixture component with index given by c. Thus, $\mathbf{z}$ may be interpreted as a coordinate on one of several local linear approximations of the manifold. The density on vectors $\mathbf{x}$ follows from the joint density over $\mathbf{x}$, $\mathbf{z}$, and c:

$$p(\mathbf{x}, \mathbf{z}, c) = p(\mathbf{x}|\mathbf{z}, c)p(\mathbf{z}|c)p(c), \tag{1}$$

$$p(\mathbf{z}|c) = \mathcal{N}(\mathbf{z}; \boldsymbol{\kappa}_c, \Sigma_c), \tag{2}$$

$$p(\mathbf{x}|c, \mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c + \boldsymbol{\Lambda}_c(\mathbf{z} - \boldsymbol{\kappa}_c), \boldsymbol{\Psi}_c), \tag{3}$$

where $\mathcal{N}(\mathbf{z}; \boldsymbol{\kappa}, \Sigma)$ denotes the Gaussian density on $\mathbf{z}$ with mean $\boldsymbol{\kappa}$ and covariance matrix $\boldsymbol{\Sigma}$. The density on $\mathbf{x}$ given c is obtained by marginalizing over $\mathbf{z}$ and given by:

$$p(\mathbf{x}|c) = \int p(\mathbf{x}|\mathbf{z}, c)p(\mathbf{z}|c)\, \mathrm{d}\mathbf{z} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Gamma}_c), \tag{4}$$

$$\boldsymbol{\Gamma}_c = \boldsymbol{\Lambda}_c \Sigma_c \boldsymbol{\Lambda}_c^\top + \boldsymbol{\Psi}_c. \tag{5}$$

Finally, the marginal density on $\mathbf{x}$ is then given by the mixture:

$$p(\mathbf{x}) = \sum_{c=1}^{C} p(c)p(\mathbf{x}|c). \tag{6}$$

The columns of the matrix $\boldsymbol{\Lambda}_c$ span the subspace of component c which has an offset $\boldsymbol{\mu}_c$ from the origin. The uncertainty in the latent coordinate $\mathbf{z}$ is mapped by $\boldsymbol{\mu}_c$ and $\boldsymbol{\Lambda}_c$ into uncertainty in $\mathbf{x}$ within the subspace. The diagonal matrix $\boldsymbol{\Psi}_c$, with positive entries, adds variance in all dimensions and makes $\boldsymbol{\Gamma}_c$ a proper, i.e., positive definite, covariance matrix.[3]

The subspace spanned by $\boldsymbol{\Lambda}_c$ provides a coordinate system which can be used to reconstruct the data from low dimensional coordinates $\mathbf{z} \in \mathbb{R}^d$, at least for data that receives high likelihood under $p(\mathbf{x}|c)$. The joint model $p(\mathbf{x}, \mathbf{z}, c)$ induces a Gaussian density on $\mathbf{z}$ given $\mathbf{x}$ and c:

$$p(\mathbf{z}|\mathbf{x}, c) = \mathcal{N}(\mathbf{z}; \mathbf{m}_c, \mathbf{V}_c^{-1}), \tag{7}$$

$$\mathbf{V}_c = \Sigma_c^{-1} + \boldsymbol{\Lambda}_c^\top \boldsymbol{\Psi}_c^{-1} \boldsymbol{\Lambda}_c, \tag{8}$$

$$\mathbf{m}_c = \boldsymbol{\kappa}_c + \mathbf{V}_c^{-1} \boldsymbol{\Lambda}_c^\top \boldsymbol{\Psi}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c), \tag{9}$$

which can be used to infer the coordinates $\mathbf{z}$ in the local subspace given a data point $\mathbf{x}$ and a mixture component c. The EM algorithm [34], [35] can be used to estimate the parameters of the mixture to find a (local) maximum of the data log-likelihood:

$$\mathcal{L} = \sum_{n=1}^{N} \log p(\mathbf{x}_n). \tag{10}$$

The inverse and determinant of $\boldsymbol{\Gamma}_c$, required to evaluate $p(\mathbf{x}|c)$, are efficiently computed as:

$$\boldsymbol{\Gamma}_c^{-1} = \boldsymbol{\Psi}_c^{-1}(\boldsymbol{\Psi}_c - \boldsymbol{\Lambda}_c \mathbf{V}_c^{-1} \boldsymbol{\Lambda}_c^\top)\boldsymbol{\Psi}_c^{-1}, \tag{11}$$

$$|\boldsymbol{\Gamma}_c| = |\boldsymbol{\Psi}_c| \times |\Sigma_c| \times |\mathbf{V}_c|. \tag{12}$$

Note that the model is "overparameterized:" $\boldsymbol{\kappa}_c$ does not appear in the likelihood $p(\mathbf{x}|c)$ and any positive definite matrix $\Sigma_c = \mathbf{U}^\top \mathbf{U} \neq \mathbf{I}$ can be absorbed in $\boldsymbol{\Lambda}_c$ by setting $\boldsymbol{\Lambda}_c \leftarrow \boldsymbol{\Lambda}_c \mathbf{U}^\top$ ($\mathbf{U}$ is the Cholesky factor of Σ_c). Therefore, the likelihood is invariant to linear transformations of the local coordinates $\mathbf{z}$; in principle, we can set $\boldsymbol{\kappa}_c = \mathbf{0}$ and $\Sigma_c = \mathbf{I}$.

2.2 Aligning Local Linear Models

Above, we defined a density model for data sampled from nonlinear manifolds. Next, we describe how we can use this model in order to recover a global parameterization of the manifold by aligning the local linear models. We define a learning criterion that combines the data log-likelihood and a

3. If the data contains outliers, robust estimates of the local subspaces can be obtained using a t-distribution noise model. See [32] for a derivation of the EM algorithm for t-distributions and [33] for an application of mixtures local linear models based on t-distributions to image manifold learning.

measure of the alignment. Therefore, the estimates of the local models can be modified to accommodate a better alignment.

In contrast, other approaches for the alignment of local models [36], [37], [38] proceed in two steps: First, a mixture of local linear models is estimated (typically, on the basis of maximum likelihood) and, second, the local models are aligned on the basis of a second criterion. Such approaches thus do not allow modification of the local models to improve alignment. In [39], a mixture of linear models is proposed, but this assumes that the global low-dimensional coordinates are known. Thus, in the latter approach, the local models can be adapted, but the global low-dimensional coordinates cannot. In the approach described below, both the local models and the estimates of the global low-dimensional coordinates of the data are iteratively re-estimated to obtain an alignment of the local models.

We assume that each data point has a unique (but unknown) global coordinate on the manifold and that, locally (i.e., in a region that is assigned high likelihood by a single mixture component), there is a linear transformation that maps the local coordinates $\mathbf{z}$ to the global coordinates on the manifold. Since the likelihood is invariant to linear transformations of the local coordinates $\mathbf{z}$, we may choose $\boldsymbol{\kappa}_c$ and Σ_c to implement this linear transformation. Thus, under these assumptions, there are settings of $\boldsymbol{\kappa}_c$ and Σ_c such that $\mathbf{z}$ corresponds to the global coordinates on the manifold.

Since the global coordinates are assumed to be unique, if a data point $\mathbf{x}$ is assigned high likelihood by two or more mixture components, then the global coordinates as predicted with $p(\mathbf{z}|\mathbf{x}, c)$ using different mixture components should be, approximately, the same. In other words: Mixture components with high posterior $p(c|\mathbf{x})$ should "agree" on the global coordinate $\mathbf{z}$ of point $\mathbf{x}$. Since $\mathbf{z}$ is not known with certainty, a formal notion of "agreement" between the mixture components needs to take into account the uncertainty in $\mathbf{z}$ as given by $p(\mathbf{z}|\mathbf{x}, c)$. The conditional density on $\mathbf{z}$ given $\mathbf{x}$ takes the form of a mixture of Gaussian densities:

$$p(\mathbf{z}|\mathbf{x}) = \sum_{c=1}^{C} p(c|\mathbf{x}) p(\mathbf{z}|\mathbf{x}, c). \tag{13}$$

If several mixture components with nonnegligible posterior $p(c|\mathbf{x})$ for a data point $\mathbf{x}$ yield quite different corresponding densities $p(\mathbf{z}|\mathbf{x}, c)$ on the global coordinates, then they do not "agree" and the mixture $p(\mathbf{z}|\mathbf{x})$ exhibits several modes. On the other hand, if all components with nonnegligible posterior $p(c|\mathbf{x})$ yield exactly the same density $p(\mathbf{z}|\mathbf{x}, c)$, then the components are in perfect agreement and the mixture $p(\mathbf{z}|\mathbf{x})$ is Gaussian and exhibits a single mode.

Our goal is to find 1) a density model for the high-dimensional data and 2) a global parameterization of the manifold from which the data was sampled. The above observations suggest that, to find the parameters $\boldsymbol{\kappa}_c$ and Σ_c such that the local coordinates on the linear subspaces correspond to the global coordinates on the manifold, we can add a penalty term to the data log-likelihood that penalizes $p(\mathbf{z}|\mathbf{x})$ that depart from a (unimodal) Gaussian distribution. The penalty term we use measures, for each data point $\mathbf{x}_n$, how much the mixture $p(\mathbf{z}|\mathbf{x}_n)$ resembles a Gaussian $q_n(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{z}_n, \Sigma_n)$, through the Kullback-Leibler (KL) divergence:

$$\mathcal{D}(q_n(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}_n)) = \int q_n(\mathbf{z}) \log \frac{q_n(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}_n)} \, \mathrm{d}\mathbf{z}. \tag{14}$$

The penalized log-likelihood objective function reads:

$$\mathcal{L}' = \sum_{n=1}^{N} \Big[\log p(\mathbf{x}_n) - \mathcal{D}(q_n(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}_n)) \Big] \tag{15}$$

$$= \sum_{n=1}^{N} \Big[\mathcal{H}(q_n(\mathbf{z})) + \int q_n(\mathbf{z}) \log p(\mathbf{x}_n, \mathbf{z}) \, \mathrm{d}\mathbf{z} \Big], \tag{16}$$

where $\mathcal{H}$ denotes the entropy of a distribution, defined as:

$$\mathcal{H}(q_n(\mathbf{z})) = -\int q_n(\mathbf{z}) \log q_n(\mathbf{z}) \, \mathrm{d}\mathbf{z}. \tag{17}$$

Note that the objective $\mathcal{L}'$ does not only depend on the parameters of the mixture model p, but also on the distributions q_n which can be regarded as probabilistic estimates of the global coordinates of the data.[4]

The "mixture of experts" (MoE) approach to classification and regression [41] is related to the method presented here. The MoE approach constructs a complex (e.g., nonlinear) classification or regression function by combining several "experts." Each expert implements a simple (e.g., linear) classification or regression function that is suitable for some subset of the data space. To determine which expert should be used to classify a new data point, a gating network is used. The gating network produces weighting factors (that are positive and sum to one) by which the predictions of the individual experts are averaged. The output of both the experts and the gating network depends on the presented data point. Analogously, in the model described above, the nonlinear mapping between high-dimensional coordinates $\mathbf{x}$ and latent coordinates $\mathbf{z}$ defined in (13) is also a weighted average of simple linear Gaussian dependencies $p(\mathbf{z}|\mathbf{x}, c)$ weighted by factors $p(c|\mathbf{x})$ that switch between the "experts." The main difference is that in this paper, we consider unsupervised learning, while MoE is used for supervised learning and parameters are found that maximize the probability of the desired output given the inputs.

2.3 Parameter Estimation

The second term in the summands of our objective function defined in (16) is an expectation of the logarithm of the mixture likelihood $p(\mathbf{x}_n, \mathbf{z}) = \sum_{c=1}^{C} p(c) p(\mathbf{x}_n, \mathbf{z}|c)$ and, as a result, $\mathcal{L}'$ exhibits local maxima that are not global maxima. Below, we consider how we can use an EM-like algorithm to find a (local) maximum of the objective $\mathcal{L}'$. For each data point, we introduce a distribution $q_n(c)$ over the mixture components, in order to bound the mixture log-likelihoods:

$$\log p(\mathbf{x}_n, \mathbf{z}) \geq \log p(\mathbf{x}_n, \mathbf{z}) - \mathcal{D}(q_n(c) \| p(c|\mathbf{x}_n, \mathbf{z})) \tag{18}$$

$$= \mathcal{H}(q_n(c)) + \sum_{c=1}^{C} q_n(c) \log p(\mathbf{x}_n, \mathbf{z}, c). \tag{19}$$

The bounds on the individual mixture log-likelihoods can be combined to bound the complete objective function:

4. It is possible to weight the penalty term by a factor different from one, yielding an objective function similar to the one used in [40] for accelerated maximum likelihood estimation. It is straightforward to derive an optimization algorithm similar to the one presented here if the penalty term is multiplied by a factor in $[0, 1]$.

$$\mathcal{L}' \geq \Phi = \sum_{n=1}^{N} \mathcal{H}(q_n(c)) + \mathcal{H}(q_n(\mathbf{z})) + \sum_{n=1}^{N} \sum_{c=1}^{C} q_n(c) \int q_n(\mathbf{z}) \log p(\mathbf{x}_n, \mathbf{z}, c) \, \mathrm{d}\mathbf{z}. \tag{20}$$

We can now iteratively increase Φ, analogous to the EM algorithm, by maximizing it in turn with respect to the parameters of $q_n(c)$, $q_n(\mathbf{z})$, and p, respectively. We call this EM-like maximization of Φ the Coordinated Factor Analysis (CFA) algorithm.

Using the abbreviations, $q_{nc} = q_n(c)$, $\mathbf{x}_{nc} = \mathbf{x}_n - \boldsymbol{\mu}_c$, and $\mathbf{z}_{nc} = \mathbf{z}_n - \boldsymbol{\kappa}_c$, we can write Φ as:

$$\Phi = \sum_{n=1}^{N} \sum_{c=1}^{C} q_{nc}[\mathcal{S}_{nc} - \mathcal{E}_{nc}], \tag{21}$$

$$\mathcal{S}_{nc} = \frac{1}{2} \log |\boldsymbol{\Sigma}_n| - \log q_{nc} + \frac{d}{2} \log(2\pi), \tag{22}$$

$$\mathcal{E}_{nc} = -\log p(c) + \frac{D+d}{2} \log(2\pi) + \frac{1}{2} \log |\boldsymbol{\Sigma}_c| + \frac{1}{2} |\boldsymbol{\Psi}_c| + \frac{1}{2} \mathrm{Tr}\left\{ \boldsymbol{\Sigma}_c (\boldsymbol{\Sigma}_n + \mathbf{z}_{nc} \mathbf{z}_{nc}^\top) \right\} + \frac{1}{2} \mathrm{Tr}\left\{ \boldsymbol{\Sigma}_n \boldsymbol{\Lambda}_c^\top \boldsymbol{\Psi}_c^{-1} \boldsymbol{\Lambda}_c \right\} + \frac{1}{2} (\mathbf{x}_{nc} - \boldsymbol{\Lambda}_c \mathbf{z}_{nc})^\top \boldsymbol{\Psi}_c^{-1} (\mathbf{x}_{nc} - \boldsymbol{\Lambda}_c \mathbf{z}_{nc}). \tag{23}$$

To maximize Φ with respect to the distributions $q_n(c)$, we equate the corresponding partial derivatives to zero and find the maximizing assignment as:

$$q_{nc} \leftarrow \left[\sum_{c'=1}^{C} e^{-\mathcal{E}_{nc'}} \right]^{-1} e^{-\mathcal{E}_{nc}}. \tag{24}$$

Similarly, to maximize Φ with respect to the $q_n(\mathbf{z})$, we set:

$$\boldsymbol{\Sigma}_n^{-1} \leftarrow \sum_{c=1}^{C} q_{nc} \mathbf{V}_c, \qquad \mathbf{z}_n \leftarrow \boldsymbol{\Sigma}_n \sum_{c=1}^{C} q_{nc} \mathbf{V}_c \mathbf{m}_{nc}.$$

Given $q_n(c)$ and $q_n(\mathbf{z})$, we maximize Φ with respect to the parameters of p. With $\tilde{q}_{nc} = q_{nc} / \sum_{n'=1}^{N} q_{n'c}$, the maximizing assignments for the mixing weights and means are:

$$p(c) \leftarrow \frac{1}{N} \sum_{n=1}^{N} q_{nc}, \quad \boldsymbol{\kappa}_c \leftarrow \sum_{n=1}^{N} \tilde{q}_{nc} \mathbf{z}_n, \quad \boldsymbol{\mu}_c \leftarrow \sum_{n=1}^{N} \tilde{q}_{nc} \mathbf{x}_n.$$

In [2], the authors suggested using a fixed-point algorithm to find the stationary points of Φ with regard to the remaining parameters of the model: $\boldsymbol{\Sigma}_c$, $\boldsymbol{\Lambda}_c$, and $\boldsymbol{\Psi}_c$. However, the parameters can be found in closed-form as shown in [42] for a constrained version of the above model and in [14] for the unconstrained model.[5] In Section 4.1, we experimentally show that the closed-form algorithm is computationally more efficient than the fixed-point algorithm, a more detailed comparison can be found in [14]. Using the weighted correlations and variances:

$$\mathbf{S}_c = \sum_{n=1}^{N} \tilde{q}_{nc} \mathbf{x}_{nc} \mathbf{z}_{nc}^\top, \qquad \mathbf{G}_c = \sum_{n=1}^{N} \tilde{q}_{nc} [\mathbf{z}_{nc} \mathbf{z}_{nc}^\top + \boldsymbol{\Sigma}_n],$$

the closed-form equations are:

$$\boldsymbol{\Sigma}_c \leftarrow \mathbf{G}_c, \qquad \boldsymbol{\Lambda}_c \leftarrow S_c \boldsymbol{\Sigma}_c^{-1},$$

$$[\boldsymbol{\Psi}_c]_{ii} \leftarrow \sum_{n=1}^{N} \tilde{q}_{nc} \Big([\mathbf{x}_{nc} - \boldsymbol{\Lambda}_c \mathbf{z}_{nc}]_i^2 + \big[\boldsymbol{\Lambda}_c \boldsymbol{\Sigma}_n \boldsymbol{\Lambda}_c^\top \big]_{ii} \Big),$$

where we used $[\cdot]_{ii}$ and $[\cdot]_i$ to denote the (i,i)th entry of a matrix and the ith entry of a vector, respectively. The computational complexity of once updating the distributions and all parameters is $O(NCDd + Nd^3))$. Unless there are fewer mixture components than latent dimensions ($C < d$), this is linear in the number of data points N, the number of mixture components C, the dimensionality of the data D, and the dimensionality of the manifold d.

2.4 Parameter Initialization

Like most other EM algorithms, the algorithm described above can terminate at poor local optima of the objective function. Careful parameter initialization, especially of the $\mathbf{z}_n$, is needed to successfully use the algorithm. In [2], the authors proposed to initialize the $\mathbf{z}_n$ by using some other unsupervised nonlinear dimension reduction technique. We can then apply the EM-like algorithm described above until convergence while keeping the $\mathbf{z}_n$ fixed at their initialized value. The $\boldsymbol{\Sigma}_n$ are also kept fixed at a small multiple of identity and the q_{nc} are initialized at random near uniform. In this manner, we learn the model on the "complete" data, i.e., each data vector $\mathbf{x}_n$ is augmented with its estimated global coordinate $\mathbf{z}_n$. After such an initialization phase, the $\mathbf{z}_n$ and $\boldsymbol{\Sigma}_n$ can be updated as well, to obtain better estimates of the global coordinates and their uncertainties. Note that the overall procedure is still unsupervised.

Other nonlinear dimension reduction methods could be used for the initialization. However, since we will use it to avoid poor local optima in the optimization procedure of the last section, the initialization should not suffer from the same problem. Therefore, the nonparametric manifold learning techniques based on minimization of convex error functions, mentioned in Section 1, are particularly useful for this purpose. Methods that first fit a mixture of local linear models [36], [37], [38] are computationally more efficient, but we found them to give suboptimal results as compared to the nonparametric methods. In our experiments described in Section 4, we use the LLE algorithm to initialize the latent coordinates, mainly because of its computational efficiency.

Finally, the number of mixture components C and the latent dimensionality d have to be set. If the number of degrees of freedom in the data is not known,[6] then d may be estimated using a variety of techniques [43], [44], [45]. Given a latent dimensionality d, the number of mixture components C can be set by employing cross-validation or a penalized log-likelihood criterion, e.g., following [46], [47]. Alternatively, a variational Bayesian learning approach [48] can be taken to estimate both the latent-dimensionality and the number of components.

3 Mapping between Manifold Embeddings

In the previous section, we presented the CFA model for nonlinear dimension reduction. In this section, we consider

5. In [2], the model was parameterized in a slightly different manner, which somewhat obscured the closed-form solution.

6. If known, d can be set to the number of degrees of freedom. However, this practice is somewhat controversial due to the existence of space-filling curves, such as Sierpinski curves and the possibility of periodic degrees of freedom, see also Section 4.4.

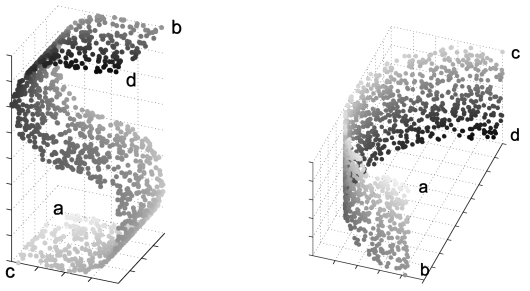


Fig. 3. Two data sets (left and right panel), the four pairs of corresponding points have been labeled "a"-"d."

how it applies to the case where multiple embeddings of the same underlying low-dimensional manifold are observed and we want to learn a mapping between these embeddings [31], [37], [49]. So, rather than one set of high-dimensional data points, we are now given two sets of high-dimensional data points: $\{\mathbf{x} \in \mathbb{R}^{D_\mathrm{x}}\}$ and $\{\mathbf{y} \in \mathbb{R}^{D_\mathrm{y}}\}$. The two sets are related through a set of correspondences: pairs $(\mathbf{x}, \mathbf{y})$ which are known to share the same low-dimensional coordinate on the manifold. For example, the two sets may be images of objects x and y viewed from different directions, or images of the face of persons x and y in different poses or with different facial expressions. A mapping between the manifolds makes it possible to map, e.g., an image sequence of person x with a varying facial expression or pose to an image sequence of person y, in such a manner that only the identity is changed from x to y but the sequence of facial expressions or poses is preserved. Such techniques could be of considerable interest for video synthesis applications.

An abstract illustration of this setting is given in Fig. 3: Two data sets are plotted respectively in the left and right panel and the correspondences are indicated by the letters "a"-"d." The goal is to predict missing correspondences, i.e., for a point in one set without a correspondence predict the coordinates of the point in the other set that would correspond to it. In other words, we consider a prediction problem where both the "inputs" and "outputs" are intrinsically low-dimensional, but specified as vectors in high-dimensional spaces.

One can also think of the data as points in the product space $\mathbb{R}^{D_\mathrm{x}} \times \mathbb{R}^{D_\mathrm{y}} = \mathbb{R}^{D_\mathrm{x}+D_\mathrm{y}}$. Corresponding pairs $(\mathbf{x}, \mathbf{y})$ can be regarded as points in the product space, the first D_x coordinates are given by $\mathbf{x}$ and the remaining D_y coordinates are given by $\mathbf{y}$. Points $\mathbf{x}$, for which we do not have a correspondence, can be regarded as points in the product space for which the first D_x coordinates are given by $\mathbf{x}$ and the other coordinates are not observed and, similarly, for points $\mathbf{y}$ without a correspondence. The goal is to estimate a density $p(\mathbf{x}, \mathbf{y})$ on the product space, which induces the conditional densities $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$ which are used to predict correspondences.

The method described in this section is related to the parameterized self-organizing map (PSOM) [50]. The PSOM also produces a mapping between two high-dimensional spaces through an underlying low-dimensional representation. The basic idea is to first find a low-dimensional representation[7] of the set of given input-output pairs $(\mathbf{x}_n, \mathbf{y}_n)$ $(n = 1, \ldots, N)$ and then to find a vector-valued function $\mathbf{f}$ that optimally reproduces each input-output pair from its low-dimensional representation. Thus, if $\mathbf{z}_n$ is the vector of low-dimensional coordinates of the nth pair, then $\mathbf{f}$ is such that, for $n = 1, \ldots, N$, we have $\mathbf{f}(\mathbf{z}_n) \approx (\mathbf{x}_n, \mathbf{y}_n)$. The function $\mathbf{f}$ is taken to be a linear combination of nonlinear basis functions. To predict the output for a new input $\mathbf{x}$, low-dimensional coordinates $\mathbf{z}^*$ are found such that $\mathbf{z}^* = \arg\min_\mathbf{z} \|\mathbf{x} - \mathbf{f}_\mathbf{x}(\mathbf{z})\|^2$, where $\mathbf{f}_\mathbf{x}$ is the function $\mathbf{f}$ restricted to the part corresponding to $\mathbf{x}$. The prediction on $\mathbf{y}$ is then given by $\mathbf{f}_y(\mathbf{z}^*)$. Compared to the CFA approach, the PSOM has two drawbacks: 1) PSOM cannot use observations without a correspondence and 2) to map between the two spaces, we have to find the low-dimensional coordinates $\mathbf{z}^*$ that optimally reproduce the input. To find $\mathbf{z}^*$, a nonconvex error function has to be minimized, which may produce erroneous results if a local but nonglobal minimum is found. Another drawback of PSOM is that, for a given input, it only produces an output where the CFA approach produces a density over possible outputs.

Another approach to the correspondence problem is to fit a MFA to the data in the product space. Note that the data has many missing values: For points without a correspondence, the first D_x or the last D_y coordinates are not observed. The EM algorithm can be used to estimate the parameters of an MFA in the presence of missing values [35], [51]. The difference with the CFA approach is that the MFA neglects the *global* manifold structure of the data; it does not relate the low-dimensional coordinate systems. Note that, if a mixture component c in a MFA does not model any correspondences (i.e., all correspondences have negligible posterior $p(c|\mathbf{x}, \mathbf{y})$ on component c), then it is not possible to (reliably) estimate the dependency between $\mathbf{x}$ and $\mathbf{y}$ for this component. On the contrary, the CFA approach estimates global low-dimensional coordinates for all data points. Therefore, even for mixture components which do not receive any posterior probability from any correspondence, the dependency between $\mathbf{x}$ and $\mathbf{y}$ can be estimated through the dependence of both $\mathbf{x}$ and $\mathbf{y}$ on the global low-dimensional coordinates $\mathbf{z}$.

In Section 3.1, we consider how, with some small changes, the CFA model also applies to this new setting. Then, in Section 3.2, we consider suitable initialization techniques.

3.1 The Probabilistic Model and Parameter Estimation

The probabilistic model and learning algorithm presented in Section 2 can be adapted to apply to the current problem. Recall that, in Section 2, we assumed the densities:

$$p(\mathbf{z}|c) = \mathcal{N}(\mathbf{z}; \boldsymbol{\kappa}_c, \boldsymbol{\Sigma}_c) \tag{25}$$

and we postulated a density $p(\mathbf{x}|\mathbf{z}, c)$, where all elements of the vector $\mathbf{x}$ are distributed independently given $\mathbf{z}$ and c. Here, we postulate a density of the same form on vectors in the product space, which can be written as:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{z}, c) = p(\mathbf{x}|\mathbf{z}, c)p(\mathbf{y}|\mathbf{z}, c), \tag{26}$$

$$p(\mathbf{x}|\mathbf{z}, c) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c^\mathbf{x} + \boldsymbol{\Lambda}_c^\mathbf{x}(\mathbf{z} - \boldsymbol{\kappa}_c), \boldsymbol{\Psi}_c^\mathbf{x}), \tag{27}$$

$$p(\mathbf{y}|\mathbf{z}, c) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_c^\mathbf{y} + \boldsymbol{\Lambda}_c^\mathbf{y}(\mathbf{z} - \boldsymbol{\kappa}_c), \boldsymbol{\Psi}_c^\mathbf{y}). \tag{28}$$

Note that the data generating model is an MFA model in the product space. The objective function Φ defined in (20) is modified to take into account only the observed data.

To facilitate notation, we introduce the binary variables o_n^x and o_n^y. If and only if $o_n^\mathrm{x} = 1$, the first D_x coordinates are observed for the nth data point, and similarly for o_n^y. Thus, only for correspondences, we have $o_n^\mathrm{x} = o_n^\mathrm{y} = 1$. We can now write the objective function as:

7. In [50], the authors suggest using self-organizing maps [11].

$$\Phi = \sum_{n=1}^{N}\sum_{c=1}^{C} q_n(c) \int q_n(\mathbf{z})[\log p(\mathbf{z},c) + o_n^{\mathbf{x}} \log p(\mathbf{x}_n|\mathbf{z},c) + o_n^{\mathbf{y}} \log p(\mathbf{y}_n|\mathbf{z},c)]\, \mathrm{d}\mathbf{z} + \sum_{n=1}^{N} \mathcal{H}(q_n(c)) + \mathcal{H}(q_n(\mathbf{z})).$$

In terms of the parameters of p, $q_n(c)$, and $q_n(\mathbf{z})$, the objective reads (ignoring constants):

$$\Phi = \sum_{n=1}^{N}\sum_{c=1}^{C} q_{nc}[\mathcal{S}_{nc} - \mathcal{E}_{nc}], \tag{29}$$

$$\mathcal{S}_{nc} = \frac{1}{2}\log|\Sigma_n| - \log q_{nc}, \tag{30}$$

$$\begin{aligned}(\mathcal{E}_{nc} = &-\log p(c) + \frac{1}{2}\log|\Sigma_c| + \frac{1}{2}\mathrm{Tr}\left\{\Sigma_c^{-1}(\Sigma_n + \mathbf{z}_{nc}\mathbf{z}_{nc}^\top)\right\} \\ &+ \frac{o_n^{\mathbf{x}}}{2}(\mathbf{x}_{nc} - \mathbf{\Lambda}_c^{\mathbf{x}}\mathbf{z}_{nc})^\top \mathbf{\Psi}_c^{\mathbf{x}^{-1}}(\mathbf{x}_{nc} - \mathbf{\Lambda}_c^{\mathbf{x}}\mathbf{z}_{nc}) \\ &+ \frac{o_n^{\mathbf{y}}}{2}(\mathbf{y}_{nc} - \mathbf{\Lambda}_c^{\mathbf{y}}\mathbf{z}_{nc})^\top \mathbf{\Psi}_c^{\mathbf{y}^{-1}}(\mathbf{y}_{nc} - \mathbf{\Lambda}_c^{\mathbf{y}}\mathbf{z}_{nc}) \\ &+ \frac{o_n^{\mathbf{x}}}{2}\left[\mathrm{Tr}\left\{\Sigma_n \mathbf{\Lambda}_c^{\mathbf{x}^\top}\mathbf{\Psi}_c^{\mathbf{x}^{-1}}\mathbf{\Lambda}_c^{\mathbf{x}}\right\} + |\mathbf{\Psi}_c^{\mathbf{x}}|\right] \\ &+ \frac{o_n^{\mathbf{y}}}{2}\left[\mathrm{Tr}\left\{\Sigma_n \mathbf{\Lambda}_c^{\mathbf{y}^\top}\mathbf{\Psi}_c^{\mathbf{y}^{-1}}\mathbf{\Lambda}_c^{\mathbf{y}}\right\} + |\mathbf{\Psi}_c^{\mathbf{y}}|\right]. \end{aligned} \tag{31}$$

For Σ_n and $\mathbf{z}_n$, the updates in the M-step become:

$$\Sigma_n^{-1} \leftarrow \sum_{c=1}^{C} q_{nc}\left[\Sigma_c^{-1} + o_n^{\mathbf{x}}\mathbf{\Lambda}_c^{\mathbf{x}^\top}\mathbf{\Psi}_c^{\mathbf{x}^{-1}}\mathbf{\Lambda}_c^{\mathbf{x}} + o_n^{\mathbf{y}}\mathbf{\Lambda}_c^{\mathbf{y}^\top}\mathbf{\Psi}_c^{\mathbf{y}^{-1}}\mathbf{\Lambda}_c^{\mathbf{y}}\right],$$

$$\begin{aligned}\mathbf{z}_n \leftarrow \Sigma_n \sum_{c=1}^{C} q_{nc}\big[&\Sigma_c^{-1}\boldsymbol{\kappa}_c + o_n^{\mathbf{x}}\mathbf{\Lambda}_c^{\mathbf{x}^\top}\mathbf{\Psi}_c^{\mathbf{x}^{-1}}(\mathbf{x}_{nc} + \mathbf{\Lambda}_c^{\mathbf{x}}\boldsymbol{\kappa}_c) \\ &+ o_n^{\mathbf{y}}\mathbf{\Lambda}_c^{\mathbf{y}^\top}\mathbf{\Psi}_c^{\mathbf{y}^{-1}}(\mathbf{y}_{nc} + \mathbf{\Lambda}_c^{\mathbf{y}}\boldsymbol{\kappa}_c)\big].\end{aligned}$$

The updates for q_{nc}, $p(c)$, $\boldsymbol{\kappa}_c$, and Σ_c remain as before, using $\tilde{q}_{nc} = q_{nc}/\sum_{n'=1}^{N} q_{n'c}$, they are:

$$q_{nc} \leftarrow \frac{e^{-\mathcal{E}_{nc}}}{\sum_{c'=1}^{C} e^{-\mathcal{E}_{nc'}}}, \quad p(c) \leftarrow \frac{1}{N}\sum_{n=1}^{N} q_{nc},$$

$$\boldsymbol{\kappa}_c \leftarrow \sum_{n=1}^{N} \tilde{q}_{nc}\mathbf{z}_n, \quad \Sigma_c \leftarrow \sum_{n} \tilde{q}_{nc}\left[\Sigma_n + \mathbf{z}_{nc}\mathbf{z}_{nc}^\top\right].$$

With $q_{nc}^{\mathbf{x}} = o_n^{\mathbf{x}} q_{nc}/\sum_{n'=1}^{N} o_{n'}^{\mathbf{x}} q_{n'c}$, $\bar{\mathbf{x}}_c^{\mathbf{x}} = \sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\mathbf{x}_{nc}$, and $\bar{\mathbf{z}}_c^{\mathbf{x}} = \sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\mathbf{z}_{nc}$, the remaining updates are:

$$\mathbf{\Lambda}_c^{\mathbf{x}} \leftarrow \left(\sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\mathbf{x}_{nc}\mathbf{z}_{nc}^\top - \bar{\mathbf{x}}_c^{\mathbf{x}}\bar{\mathbf{z}}_c^{\mathbf{x}\top}\right) \times \left(\sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\left[\Sigma_n + \mathbf{z}_{nc}\mathbf{z}_{nc}^\top\right] - \bar{\mathbf{z}}_c^{\mathbf{x}}\bar{\mathbf{z}}_c^{\mathbf{x}\top}\right)^{-1},$$

$$\boldsymbol{\mu}_c^{\mathbf{x}} \leftarrow \sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\mathbf{x}_n - \mathbf{\Lambda}_c^{\mathbf{x}}\bar{\mathbf{z}}_c^{\mathbf{x}},$$

$$\left[\mathbf{\Psi}_c^{\mathbf{x}}\right]_{ii} \leftarrow \sum_{n=1}^{N} q_{nc}^{\mathbf{x}}\left(\left[\mathbf{x}_{nc} - \mathbf{\Lambda}_c^{\mathbf{x}}\mathbf{z}_{nc}\right]_i^2 + \left[\mathbf{\Lambda}_c^{\mathbf{x}}\Sigma_n\mathbf{\Lambda}_c^{\mathbf{x}\top}\right]_{ii}\right).$$

The updates for $\boldsymbol{\mu}_c^{\mathbf{y}}$, $\mathbf{\Lambda}_c^{\mathbf{y}}$, and $\mathbf{\Psi}_c^{\mathbf{y}}$ are analogous. Note that the above is straightforwardly generalized to settings with more than two sets of points.

3.2 Parameter Initialization

As before, proper initialization of the $\mathbf{z}_n$ is crucial in order to find good parameter estimates with our EM-like algorithm. In the current setting, we cannot directly apply the LLE-based parameter initialization used before because the missing values do not allow us to compute nearest neighbors and reconstruction weights in the product space.

In [31], an approach was proposed to extend the LLE algorithm to the current setting. The idea is to perform LLE on both sets of points simultaneously, and to constrain the latent coordinates of corresponding points to be identical. The error function is defined as the sum of the LLE error function for both sets of points. The error function is quadratic and can be minimized, subject to the correspondence constraints, by computing the $(d+1)$ eigenvectors with smallest eigenvalues of a sparse matrix.

4 Experimental Results

In this section, we present experimental results obtained with the approaches described in the previous sections. In Section 4.1, we present qualitative results of applying the CFA algorithm to data sampled from a single manifold. We compare CFA performance with that of several alternative approaches in Section 4.2, and in Section 4.3, we compare our closed-form CFA algorithm with the fixed-point algorithm of Roweis et al. In Section 4.4, we use the CFA model to predict correspondences between two manifolds and compare its performance with that obtained using an MFA model.

4.1 Manifold Modeling with CFA: Qualitative Results

The CFA model can be used in two directions: First, the model can be used for dimension reduction, i.e., mapping a point $\mathbf{x}$ in the data space to a point $\mathbf{z}$ in the latent space. Second, the model can be used for data reconstruction, i.e., mapping a point $\mathbf{z}$ in the latent space to a point $\mathbf{x}$ in the data space. The mappings can be performed by evaluating $\mathbf{E}_{p(\mathbf{z}|\mathbf{x})}[\mathbf{z}]$ and $\mathbf{E}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$.[8]

In Fig. 4, we illustrate the two directions in which we can use the model using the "S" data set of Fig. 2. We initialized the latent coordinates with the LLE algorithm using 10 nearest neighbors and we used a mixture of $C = 10$ components (below, we consider the effect of C on the results). In Fig. 4a, the original three-dimensional data is depicted on which the model was trained. Fig. 4b depicts the two-dimensional latent representation of the original data and a rectangular grid in the same space. Fig. 4a also depicts the grid when mapped to the data space with the trained model. The illustration shows that the nonlinear subspace containing the data was indeed recovered from the training data and that accurate nonlinear mapping between the data space and latent space is possible using $C = 10$ components.

Since, for the "S" data set of Fig. 2, the true latent coordinates are available, we can measure how well they are recovered by CFA and LLE, repsectively. We applied the LLE algorithm, and, subsequently, the CFA algorithm, on a

8. It is an interesting direction for further research to consider alternatives to average over the local projections. For example, it might be possible to use notions like the Karger mean [52], or methods used in [27] for interpolation on nonlinear manifolds.

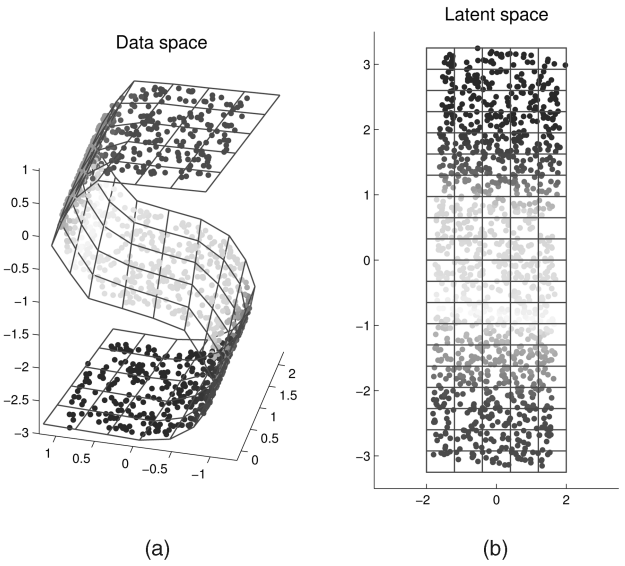


Fig. 4. (a) Original data in $\mathbb{R}^3$ and a grid mapped from the latent space to the data space with the model. (b) Original grid and data mapped to latent space.

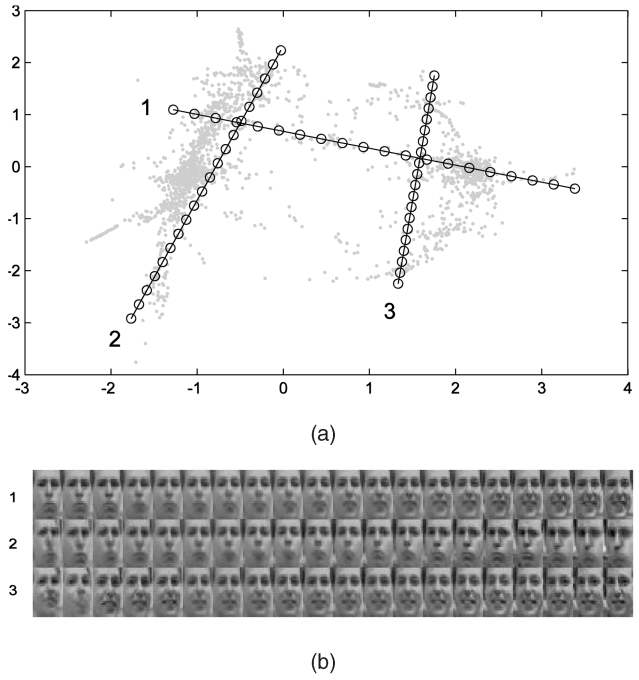


Fig. 5. (a) Recovered latent representation of the images. (b) Images generated with model along the line segments shown in the top panel.

random subset that contains 80 percent of the available 1,240 data points. We then used the CFA model to map the remaining 20 percent to the latent space. For LLE, the remaining points were mapped to the latent space by the nonparametric method described in [39]. Since both CFA and LLE find the latent representation only upto a linear transformation, we first optimally linearly transform the recovered latent coordinates before we measure their squared distance to the known true latent coordinates. We call the sum of these squared distances the "embedding error." For LLE, we used $k = 6, \ldots, 20$ neighbors and we used the LLE results to learn CFA models with $C = 6, \ldots, 20$ mixture components. For all combinations of k and C, we found that the average (over 10 random selections of test and train data) difference in embedding error for LLE and CFA is very small: about two orders of magnitude smaller than the standard deviations in the embedding errors.

We repeated this experiment using the data shown in Fig. 1; here, the two-dimensional manifold (the true latent coordinates are given by the horizonal and vertical position of the square, which both range from 1 to 20) is nonlinearly embedded in a 841-dimensional space. Here, we used for LLE $k = 5, 10, \ldots 45, 50$ neighbors and $C = 10, 20, \ldots, 60, 70$ mixture components for CFA. Again, except for a few combinations, the average difference in the embedding error is smaller than the standard deviation in the embedding errors (although the differences and standard deviations are of the same order of magnitude for this data set). Both LLE and CFA make, on average, an error of about one pixel in mapping the image to the position of the square.

Next, we show results obtained when applying the model to a data set of facial images, which vary in the pose and expression of the face. This data set consists of 1,965 images of $20 \times 28 = 560$ pixels each. The data set was also used in [2] where the authors showed similar results as shown here (although the results here are obtained with the faster closed-form solution algorithm).[9] For initialization, we used the LLE algorithm with 14 nearest neighbors.

The recovered latent representation of the data is shown in Fig. 5, where each dot represents the (expected) coordinates $\mathbf{z}$ of an image (we use the model to map the high-dimensional images to the two-dimensional latent space). We used the trained model also to generate images $\mathbf{x}$ from latent coordinates $\mathbf{z}$ along three straight trajectories in the latent space (the model is used to map two-dimensional coordinates along the trajectories to 560-dimensional images). The trajectories are plotted in the figure and the corresponding generated images are shown in Fig. 5b. The latent coordinates form two clusters; the first trajectory passes through both clusters and the other two trajectories stay within a single cluster. From the generated images along the trajectories, we can see that the clusters roughly correspond to images of smiling and nonsmiling faces. The reconstructed images along the second and third line segment show that the variation within the clusters corresponds to pose variation of the faces.

4.2 Quantitative Comparisons

We compared CFA with a probabilistic variant of self-organizing maps called self-organizing mixture models (SOMM) [12] and generative topographic mapping (GTM) [13]. As CFA, these are probabilistic methods that allow for a mapping between the data space and latent space in both directions. Therefore, it is possible to make a direct comparison between the generalization performance in terms of data log-likelihood and reconstruction errors. To measure the generalization performance of these methods,

9. The data is available at http://www.cs.toronto.edu/~roweis.

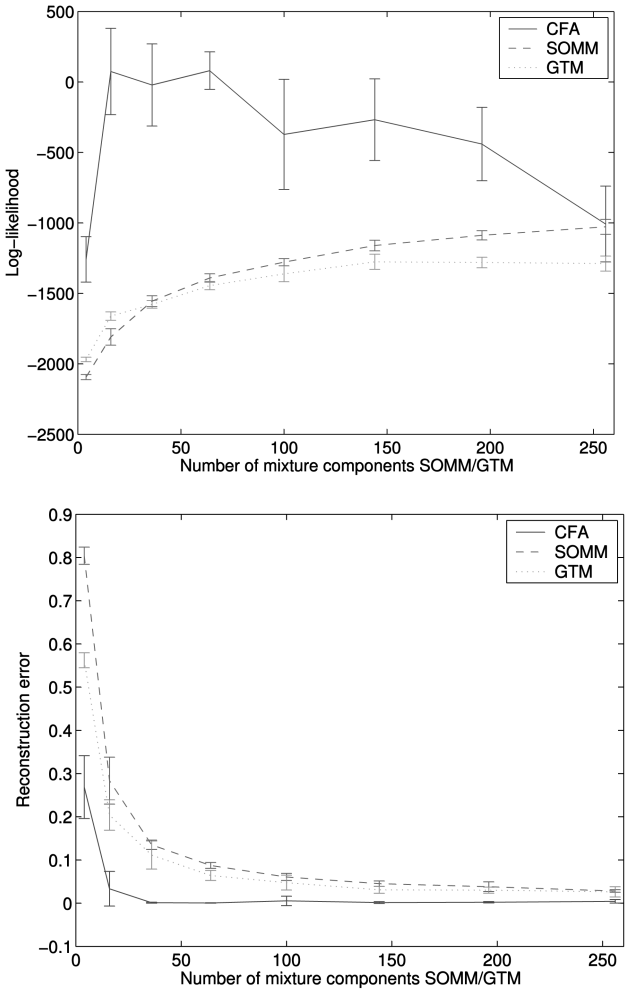


Fig. 6. Comparison of CFA, GTM, and SOMM using synthetic data.

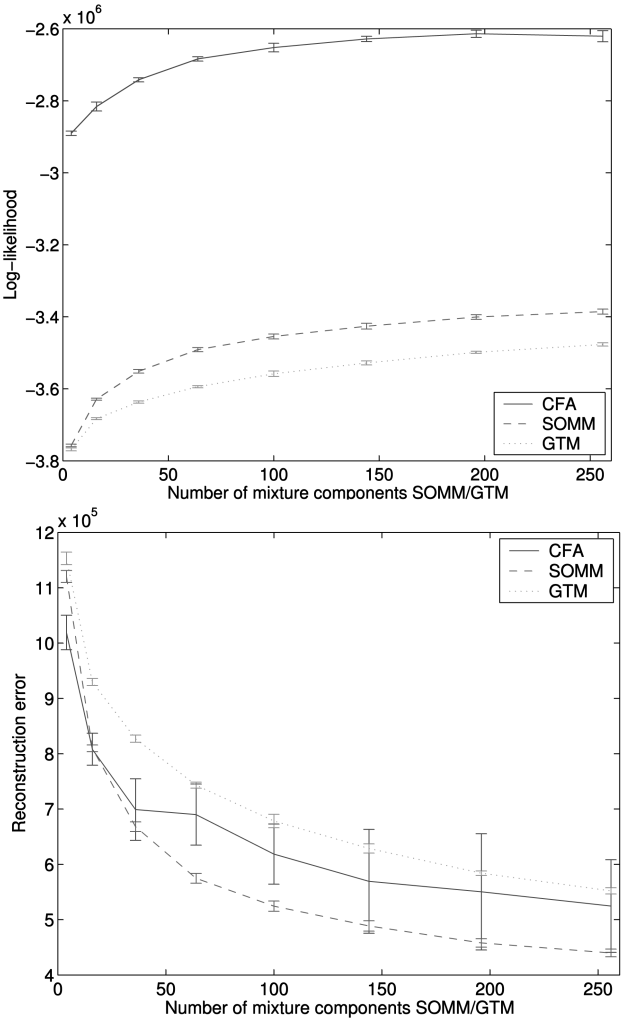


Fig. 7. Comparison of CFA, GTM, and SOMM, results for image data.

we use a separate set of test data not included when estimating the model parameters. The reconstruction $\hat{\mathbf{x}}_n$ of a point $\mathbf{x}_n$ using the model is obtained by mapping $\mathbf{x}_n$ to a single point $\mathbf{z}_n$ in the latent space and then mapping $\mathbf{z}_n$ back to the data space.[10] The reconstruction error is defined as:

$$E_{rec} = \sum_n \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2.$$

All models used a two-dimensional latent space; for GTM and SOMM, the nodes were placed at a square rectangular grid in the latent space such that the node locations had zero mean and identity covariance. For GTM and SOMM, we use Gaussian component densities with isotropic covariance matrix. For GTM, we used as many basis functions as mixture components. The basis functions were of the form $\phi_s(\mathbf{z}) = \exp(-\|\mathbf{z} - \boldsymbol{\kappa}_s\|^2/2\sigma^2)$, and we used $\sigma^2 = 1/10$; which yielded the best results of several values that were tried. For CFA, we constrained the noise variance matrices $\boldsymbol{\Psi}_c$ to be of the form $\boldsymbol{\Psi}_c = \sigma_c^2\mathbf{I}$ so that, also here, the variance outside the subspaces is isotropic. Note that the CFA model has $(d+1)$ times more parameters per mixture component than the SOMM and GTM models to encode $\boldsymbol{\Lambda}_c$. To obtain a fair comparison, we compare models with an equal number of parameters. For SOMM and GTM, we used $C \in \{2^2, 4^2, 6^2, \ldots, 16^2\}$; for CFA, the number of components was taken to be $C' = \lfloor C/(d+1) \rfloor$.

The comparison is based on two data sets: the synthetic "S" data set used before (using a training and test set of 600 points) and a data set of 2,000 images of 40×40 pixels each of a face looking in different directions (using 1,500 training images and 500 test images).[11] The reported results are averages and standard deviations over 10 randomly drawn train and test sets. Results obtained using the synthetic data set are summarized in Fig. 6, and those obtained using the image data set in Fig. 7. In Fig. 8, we also plotted some of the reconstructions obtained for the second data set using the different models (using $C = 256$ components for SOMM and GTM and $C' = 85$ components for CFA). As expected from the results in Fig. 7, the reconstructions of the different methods appear quite similar.

The results lead to the following conclusions. The CFA model is able to achieve significantly higher log-likelihood on the test data than the SOMM and GTM

10. For GTM, we map latent coordinates to the data space using the generalized linear model. For SOMM, we assign the high-dimensional coordinates of the mixture component with nearest mean in the latent space.

11. The data is available at http://lear.inrialpes.fr/~verbeek.

Fig. 8. From top to bottom: original images, and reconstructions using CFA, SOMM, and GTM.

models, especially when a moderate model capacity is used (small number of mixture components/parameters). Compared to log-likelihood, the differences in reconstruction errors are relatively small. The discrepancy between the reconstruction and log-likelihood scores can be understood as follows: SOMM and GTM are based on mixtures where the components are Gaussians with isotropic covariance. The isotropic noise makes SOMM and GTM have to "leak" a considerable amount of mass to parts of the data space that are not filled by the low-dimensional manifold, even though they use about three times more mixture components. The CFA model, on the other hand, can locally concentrate its mass in the linear subspace approximating the manifold. The reconstruction error measures the squared Euclidean distance between a data point and its reconstruction. Note that the reconstruction is the mean of the distribution on $\mathbf{x}$ given its low-dimensional representation $\mathbf{z}$; the reconstruction error does not take into account the uncertainty in this distribution.

The standard deviation in the results obtained with the CFA model are generally larger or comparable to those obtained for SOMM and GTM. This is caused by the sensitivity to the initialization of the CFA model. This sensitivity might be resolved by using more robust initialization techniques, e.g., it is possible to minimize the sum of the LLE error function using several numbers of neighbors or to use bootstrap-like procedures as in [31].

Using many mixture components, CFA shows overfitting on both data sets in terms of log-likelihood. This effect is well known [53] and is caused by the fact that using many components more parameters need to be estimated. However, since the amount of available data is limited, the parameters cannot be accurately estimated and, as a result, the model is erroneous. The overfitting effect is not observed for SOMM, and for GTM only on the synthetic data set. This is due to the fact that the latter two models use an isotropic noise model.

4.3 Comparison of Closed-Form and Fixed-Point Algorithm

Next, we compare the EM algorithm using our closed-form M-step and using the fixed-point M-step proposed in [2]. The complexity of both M-steps is $O(Dd^2)$. Thus, the factor of speed-up depends on the number of iterations needed by the iterative approach to reach a fixed-point. We measured the performance when applying the algorithms to two data sets. The first one is the "S" data set of Fig. 2. The second data set is a set of 698 computer generated images of 64×64 pixel each of a face seen from different directions and under different lighting conditions.[12] To speed-up experimentation, the images were first projected on the 30-dimensional principal components subspace which contains over 90 percent of the total data variance.

To start the fixed-point iterations, we initialized the parameters at their value found in the previous M-step. We considered the fixed-point algorithm to have converged if the maximum relative change in the parameters dropped below 10^{-4}. If the fixed-point algorithm did not converge in 1,000 iterations, it was terminated and the parameter values of the previous M-step were retained. To quantify the speed-up of the closed-form equations in practice and to see whether they lead to better final parameter estimates, we measured: 1) runtime of the algorithm,[13] 2) value of $\mathcal{L}'$ attained after convergence, and 3) number of EM steps required to converge.

The results are summarized in Table 1 by averages and standard deviations from 10 runs. The results show that the closed-form algorithm is significantly faster and yields higher values for the objective function in a comparable number of EM iterations. We conclude that the closed-form update equations are to be preferred over the fixed-point algorithm.

4.4 Mapping between Multiple Embeddings of a Manifold

In this section, we experimentally compare our CFA approach against an MFA to predict high-dimensional correspondences. We show how the performance of the approaches depends on the amount of correspondences and the number of mixture components. First, we compare prediction accuracy on two synthetic data sets to gain insight in the differences between the two approaches. Then, we compare them using a data set consisting of gray scale images of two objects to determine if the differences observed with the synthetic data sets are also apparent in more realistic data sets.

In the first experiment, we used the data set shown in Fig. 3. Both sets contain 1,240 points in $\mathbb{R}^3$; in each

12. This data set is available online at http://isomap.stanford.edu.

13. To measure convergence of the EM algorithm, we checked whether the relative change in $\mathcal{L}'$ was smaller than 10^{-4}.

TABLE 1
Comparison between Closed-Form and Fixed-Point Algorithms

Results for the 'S' data set, $d = 2, C = 10$**.**			
	# EM steps	time	Φ
closed-form	123.5 ± 49.5	6.6 ± 3.9	0.9 ± 0.6
fixed-point	125.5 ± 33.8	77.8 ± 29.3	0.2 ± 0.8
Results for the isomap face data set, $d = 3, C = 50$**.**			
	# EM steps	time	Φ
closed-form	78.4 ± 7.0	30.0 ± 2.8	9.7 ± 0.7
fixed-point	76.7 ± 9.1	198.0 ± 32.7	8.1 ± 1.1

TABLE 2
Reconstruction Errors for the Synthetic Data Set

C	2	3	4	5	6	7	8
5%	**0.27**	**0.34**	**0.34**	**0.22**	**0.25**	**0.25**	**0.23**
	0.74	**0.98**	**1.26**	**1.13**	**1.45**	**1.40**	**1.28**
15%	**0.18**	0.21	**0.14**	**0.15**	**0.13**	**0.09**	**0.09**
	0.34	0.33	**0.57**	**0.51**	**0.71**	**0.67**	**0.67**
25%	0.16	**0.13**	**0.12**	**0.08**	**0.08**	**0.09**	**0.06**
	0.18	**0.19**	**0.25**	**0.30**	**0.37**	**0.40**	**0.42**
35%	**0.16**	0.13	**0.12**	**0.09**	**0.08**	**0.09**	**0.06**
	0.18	0.18	**0.16**	**0.20**	**0.23**	**0.23**	**0.26**
45%	0.16	**0.13**	**0.11**	**0.10**	**0.07**	**0.06**	**0.05**
	0.17	**0.15**	**0.17**	**0.18**	**0.18**	**0.20**	**0.19**
55%	0.16	**0.13**	**0.10**	**0.08**	**0.06**	**0.05**	**0.05**
	0.16	**0.15**	**0.14**	**0.16**	**0.16**	**0.14**	**0.13**

For each percentage, CFA results are printed above MFA results.

experiment, we used a random subset of 600 points from each set to fit the models and a second set of 600 corresponding points to assess the quality of the fitted models. The results reported below are averages over 20 experiments. We trained CFA and MFA models using latent-dimensionality of $d = 2$, while varying the percentage of training data for which correspondences are available (ranging from 5 percent up to 55 percent with 10 percent intervals, i.e., using from 30 up to 330 correspondences) and the number C of mixture components (ranging from $C = 2$ to $C = 8$). To assess the quality of the fitted models, we used the test data, for which all six coordinates are known. For the test data, we predicted the last three coordinates given the first three coordinates with the trained models and vice versa.

When using an MFA, each mixture component c is a Gaussian density on the product space which induces the conditional Gaussian $p(\mathbf{y}|\mathbf{x}, c)$. Combining the different components, we obtain the conditional (mixture) density $p(\mathbf{y}|\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{y}|\mathbf{x}, c)p(c|\mathbf{x})$. Hence, the expected value of $\mathbf{y}$ under this distribution, which we denote by $\hat{\mathbf{y}}$, is the average of the means of $p(\mathbf{y}|\mathbf{x}, c)$, where each mean is weighted by the corresponding posterior probability $p(c|\mathbf{x})$. The same holds for the CFA models and the predictions on $\mathbf{x}$ given $\mathbf{y}$.

As an error measure of the models, we used the squared difference between the predicted coordinates $\hat{\mathbf{x}}$ (or $\hat{\mathbf{y}}$) and the true value of the coordinates of $\mathbf{x}$ (or $\mathbf{y}$), averaged over all test data points and over the coordinates. Thus, for the $N = 600$ test points, we measure:

$$E_{rec} = \frac{1}{ND_{\mathbf{x}}} \sum_{n=1}^{N} \|\hat{\mathbf{x}}_n - \mathbf{x}_n\| + \frac{1}{ND_{\mathbf{y}}} \sum_{n=1}^{N} \|\hat{\mathbf{y}}_n - \mathbf{y}_n\|.$$

In Table 2, we tabulated, for different amounts of correspondences and mixture components, the error E_{rec} obtained with CFA and MFA models. Results where the CFA error is significantly[14] smaller are printed bold.

The results show that if many correspondences are available, CFA and MFA give comparable results and if only a few correspondences are given, CFA performs significantly better than MFA. This difference is more pronounced as the number of mixture components becomes larger. The explanation of the effect is that, if the average number of correspondences per mixture component becomes small, then in a MFA model, there can be mixture components which have (almost) no responsibility for correspondences (i.e., observations without missing values). In such cases, it is not possible to determine the dependencies between coordinates in the two different spaces since, for each data point with nonnegligible responsibility for the component, either only the first D_1 coordinates are observed or only the last D_2 coordinates are. In comparison, the CFA model can exploit the incomplete observations to determine these dependencies; this is possible through the dependencies between the incomplete observations and the estimated global low-dimensional coordinates on the manifold.

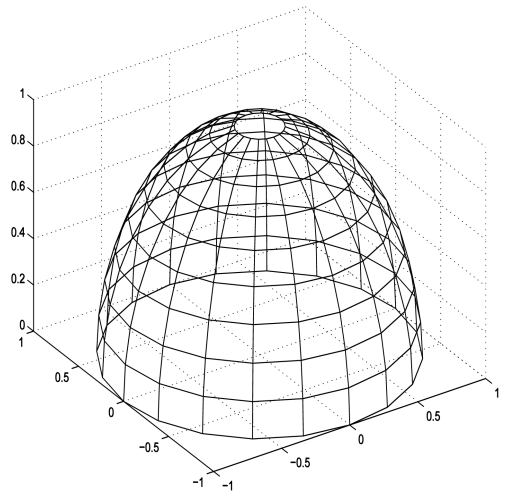
In the second experiment, we also compare the reconstruction errors using MFA and CFA models, but we use a more realistic data set of much higher-dimensionality. The data consists of 2,500 gray-scale images of 64×64 pixels of each of two toy puppets as viewed from different directions. In Fig. 9, some corresponding views of the two puppets are depicted. The images, originally used in [54], were provided by Peters et al., who recorded them at the Institute for Neural Computation of the Ruhr-Universität-Bochum, Germany. Images of the objects were recorded while moving the camera over the *viewing hemisphere*, see Fig. 10a. The viewing hemisphere was sampled at $3.6°$ intervals in longitude (yielding 100 steps to complete the circle) and at $3.6°$ intervals in latitude (yielding 25 steps to go from the equator of the hemisphere to the pole). Note that the images recorded near the top of the hemisphere differ considerably since these are different rotations of the top view of the object.

There are only two degrees of freedom in each set of images since the images are determined by the position of the

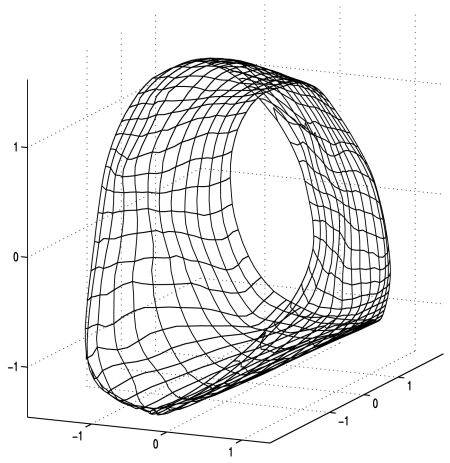
14. To determine significance, we used a t-test with 19 DOF and $p = 0.05$.

Fig. 9. Examples of the images of the two toy puppets: The top row contains images of the cat figure and the bottom row images of the dwarf figure. Corresponding views are displayed above each other.

camera, which is, in turn, determined by its longitude and latitude on the hemisphere. Since the longitude is a periodic degree of freedom, the images can be embedded on the surface of a cylinder in a Euclidean space. In principle, the images can also be embedded, while preserving nearest neighbor relations, in a two-dimensional space by embedding images with equal latitude on concentric circles with a radius that is monotonically increasing with the latitude.



(a)



(b)

Fig. 10. The viewing hemisphere: (a) The object was fixed in the center of the sphere and images were recorded when the camera was placed at different locations on the hemisphere and directed toward the object. (b) Three-dimensional embedding of 625 of the 2,500 images.

However, such a two-dimensional embedding is not returned by the LLE-based algorithm since it is not directly aiming at the preservation of nearest neighbor relations. Three dimensions are required to recover a cylinder-like embedding of the images in which only similar images are embedded nearby. Therefore, we used a three-dimensional latent space for the CFA models, and to obtain a fair comparison, we also use $d = 3$ for the MFA models. In Fig. 10, we illustrated the three-dimensional latent representation recovered by the LLE-based initialization method when using all 2,500 images of each set and 125 correspondences. The recovered coordinates indeed trace-out a cylinder-like shape.

In order to speed-up the experimentation, the images were first projected to the 100-dimensional linear PCA subspace of the $64 \times 64 = 4,096$-dimensional space spanned by the pixel values. Over 90 percent of the original variance in the data was contained in this subspace with approximately 40 times fewer dimensions. Since the discarded dimensions contain only a small fraction of the data variance, projecting the original data to the PCA subspace is expected to have little effect on the obtained results. Below, we also show results obtained when using 1,000-dimensional PCA projection. We trained CFA and MFA models with a number of mixture components ranging from $C = 10$ to $C = 60$ with steps of 10. The number of images with a correspondence was set to 2, 5, 10, 20, and 50 percent. Of each object, 2,000 images were used for training and 500 to assess the reconstruction error E_{rec} from (32).

The obtained errors, averaged over six random selections of training and test data, are tabulated in Table 3. Statistically significant differences are printed bold (using a t-test with 5 degrees of freedom and $p = 0.05$). In Fig. 11, we plotted averages of the error of CFA models and MFA models using $C = 40$ components against the percentage of correspondences that was used. From this figure, it can be clearly observed that, already, with relatively few correspondences, CFA models make accurate predictions and MFA models require many more correspondences to obtain similar errors.

To compare MFA and CFA qualitatively, we plotted the true and predicted correspondence for some of the test examples in Fig. 12. For reference, we also included reconstructions of the true correspondences obtained with a three dimensional linear PCA projection.[15] The depicted results were obtained by training models with $C = 65$ mixture

15. The PCA results were obtained by performing PCA on both sets of images separately and reconstructing the *true* correspondence from its three dimensional PCA representation. This is the best possible result that could be obtained using a linear reconstruction of the images from a three-dimensional representation.

TABLE 3
Reconstruction Errors for the Image Data Set

C	1	10	20	30	40	50	60
2%	**4.78**	**2.31**	**1.44**	**1.15**	**1.51**	**2.32**	**1.49**
	6.47	**8.16**	**7.21**	**7.14**	**7.06**	**7.94**	**7.36**
5%	**4.64**	**1.99**	**1.18**	**0.96**	**0.74**	**0.73**	**0.55**
	6.47	**5.25**	**4.89**	**4.71**	**4.20**	**4.30**	**4.55**
10%	**4.21**	**1.78**	**1.08**	**0.84**	**0.68**	**0.54**	**0.59**
	6.42	**3.74**	**4.29**	**3.35**	**2.71**	**3.27**	**2.95**
20%	**4.04**	**1.54**	**0.93**	**0.73**	**0.66**	**0.50**	**0.46**
	6.42	**2.51**	**2.55**	**2.40**	**2.26**	**2.34**	**1.78**
50%	**3.89**	**1.47**	0.84	**0.64**	**0.64**	**0.44**	**0.36**
	6.32	**1.74**	1.02	**1.07**	**0.86**	**0.72**	**0.89**

For each percentage, CFA results are printed above MFA results.

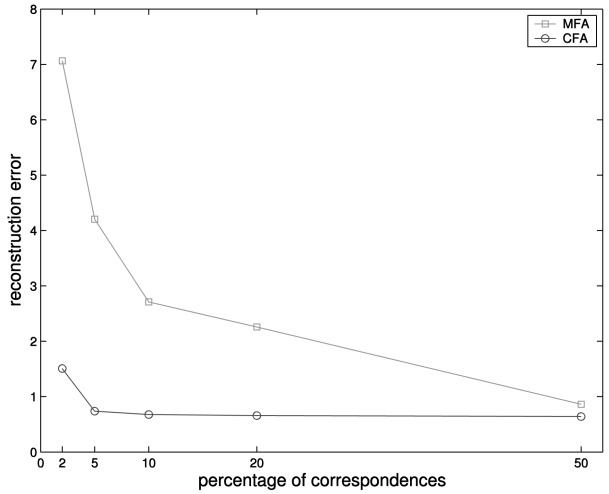


Fig. 11. Average reconstruction errors obtained with CFA and MFA models using 40 components.

components on 2,000 images of each object with 100 images in correspondence. Before fitting the models, the images in each set were projected on the first 1,000 principal components derived from the images in that set, preserving over 99.6 percent of the variance in each set. The average errors (per image and per dimension) were 0.127 for CFA and 0.732 for MFA. Clearly, the CFA model—which exploits the global manifold structure of the data—yields superior predictions as compared to those obtained with the MFA model.

5 Conclusions

In this paper, we considered the Coordinated Factor Analysis (CFA) approach to combine linear models to form a nonlinear model. Compared to GTM and SOMM, the advantage of this method is that the low-dimensional representation is not restricted to a discrete set of points (the nodes in SOMM and

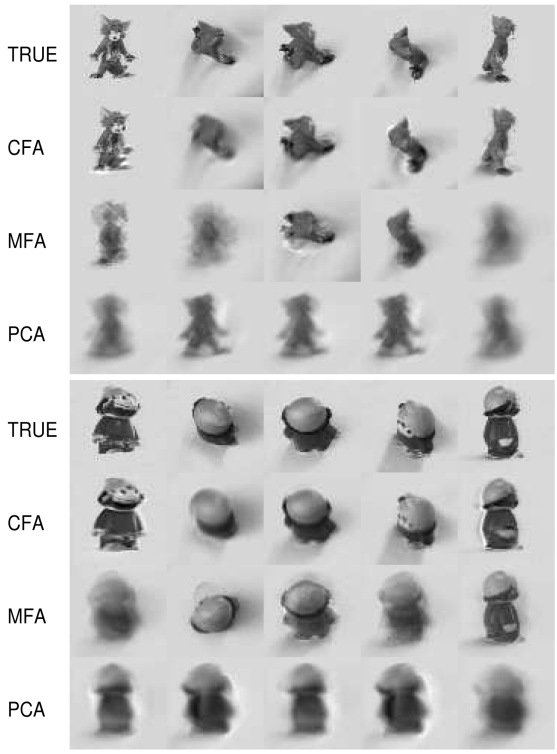


Fig. 12. The top panel shows the (predicted) corresponding view of the cat given a view of the dwarf and the bottom panel shows (predicted) corresponding view of the dwarf given a view of the cat. In each panel, from top to bottom: true correspondence, reconstruction with CFA, MFA, and 3D PCA projection.

GTM), but offers a continuous low-dimensional representation. Furthermore, the conditional density $p(\mathbf{z}|\mathbf{x})$ on latent coordinates given a data vector (and, vice versa, $p(\mathbf{x}|\mathbf{z})$) is a Gaussian mixture density whose parameters and expectation are readily computed. We presented an improved parameter estimation scheme, as compared to that proposed in [2], which replaces a fixed-point iteration within the M-step of the EM-like algorithm by a closed-form solution. The experimental results presented in Section 4 indicated that the CFA approach compares favorably to SOMM and GTM in the sense that fewer mixture components are needed to obtain similar reconstruction errors (errors in predicting the high-dimensional points from their low dimensional representation).

In Section 3, we generalized the CFA approach to a setting where the goal is to find a model that enables us to predict between two high-dimensional variables. In this setting, the training data consists of two sets of high-dimensional points. For some points, the corresponding point in the other set is given. We compared the CFA approach with a mixture of factor analyzers approach, and experimentally found that when only a few correspondences are given, the CFA approach leads to a significantly better prediction accuracy, both for synthetic and natural data. The difference in performance is explained by the fact that CFA successfully uses the manifold structure of the data to determine the dependencies between the high-dimensional variables, even if few correspondences are given.

A line of research we want to pursue in the future is the use of nonparametric semisupervised learning techniques, in combination with the models presented here for regression problems. In semisupervised learning, for some images, the response variables are known and are used to infer a density on the response variables of the other images. The response variables can be either the DOF that span the image manifold or some smooth function on the manifold (e.g., the absolute angular deviation from the frontal pose of a face). In this manner, the unsupervised data can be used to bias the learning of the regression function. This is similar to the way in which unsupervised data was used for correspondence learning here.

ACKNOWLEDGMENTS

The author would like to thank the reviewers for their detailed comments and suggestions. The author is also indebted to Nikos Vlassis and Ben Kröse for many useful discussions. This research was carried out when the author was with the University of Amsterdam, currently he is with INRIA Rhone-Alpes

REFERENCES

[1] A. Leonardis and H. Bischof, "Kernel and Subspace Methods for Computer Vision," *Pattern Recognition,* vol. 36, no. 9, pp. 1925-1927, 2003.

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

[3] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. Optical Soc. Am. A,* vol. 4, no. 3, pp. 519-524, 1987.

[4] I.T. Jolliffe, *Principal Component Analysis.* New York: Spinger-Verlag, 1986.

[5] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[6] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *Int'l J. Computer Vision,* vol. 14, pp. 5-24, 1995.

[7] E. Oja, "Data Compression, Feature Extraction, and Autoassociation in Feedforward Neural Networks," *Proc. Int'l Conf. Artificial Neural Networks,* pp. 737-745, 1991.

[8] T. Hastie and W. Stuetzle, "Principal Curves," *J. Am. Statistical Assoc.,* vol. 84, no. 406, pp. 502-516, 1989.

[9] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Learning and Design of Principal Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 3, pp. 281-297, Mar. 2000.

[10] K. Chang and J. Ghosh, "A Unified Model for Probabilistic Principal Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 1, pp. 22-41, Jan. 2001.

[11] T. Kohonen, *Self-Organizing Maps.* New York: Spinger-Verlag, 1995.

[12] J.J. Verbeek, N. Vlassis, and B.J.A. Kröse, "Self-Organizing Mixture Models," *Neurocomputing,* vol. 63, pp. 99-123, 2005.

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

[14] J.J. Verbeek, "Mixture Models for Clustering and Dimension Reduction," PhD dissertation, Univ. of Amsterdam, 2004.

[15] I.K. Fodor, "A Survey of Dimension Reduction Techniques," Technical Report UCRL-ID-148494, Lawrence Livermore Nat'l Laboratory, Center for Applied Scientific Computing, 2002.

[16] M.Á. Carreira-Perpiñán, "A Review of Dimension Reduction Techniques," Technical Report CS-96-09, Dept. of Computer Science, Univ. of Sheffield, 1997.

[17] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear-Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.

[18] S.T. Roweis and L.K. Saul, "Nonlinear-Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.

[19] 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.

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

[21] X. He and P. Niyogi, "Locality Preserving Projections," *Advances in Neural Information Processing Systems,* vol. 16, pp. 153-160, 2004.

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

[23] 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.

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

[25] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le 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. 177-184, 2004.

[26] N. Kambhatla and T.K. Leen, "Fast Nonlinear Dimension Reduction," *Advances in Neural Information Processing Systems,* vol. 6, pp. 152-159, 1994.

[27] C. Bregler and S.M. Omohundro, "Nonlinear Image Interpolation Using Manifold Learning," *Advances in Neural Information Processing Systems,* vol. 7, pp. 973-980, 1995.

[28] M.E. Tipping and C.M. Bishop, "Mixtures of Probabilistic Principal Component Analysers," *Neural Computation,* vol. 11, no. 2, pp. 443-482, 1999.

[29] G.E. Hinton, P. Dayan, and M. Revow, "Modeling the Manifolds of Images of Handwritten Digits," *IEEE Trans. Neural Networks,* vol. 8, no. 1, pp. 65-74, 1997.

[30] R. Bellman, *Adaptive Control Processes: A Guided Tour.* Princeton Univ. Press, 1961.

[31] J.H. Ham, D.D. Lee, and L.K. Saul, "Learning High-Dimensional Correspondences from Low-Dimensional Manifolds," *Proc. Workshop the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining,* 2003.

[32] X. Meng and D. van Dyk, "The EM Algorithm—An Old Folk Song Sung to a Fast New Tune," *J. Royal Statistical Soc., Series B (Methodological),* vol. 59, no. 1, pp. 511-567, 1997.

[33] D. de Ridder and V. Franc, "Robust Subspace Mixture Models Using t-Distributions," *Proc. British Machine Vision Conf.,* pp. 319-328, 2003.

[34] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Series B (Methodological),* vol. 39, no. 1, pp. 1-38, 1977.

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

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

[37] J.J. Verbeek, S.T. Roweis, and N. Vlassis, "Nonlinear CCA and PCA by Alignment of Local Models," *Advances in Neural Information Processing Systems,* vol. 16, pp. 297-304, 2004.

[38] J. Wieghardt, "Learning the Topology of Views: From Images to Objects," PhD dissertation, Ruhr-Univ.-Bochum, Bochum, Germany, 2001.

[39] 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.

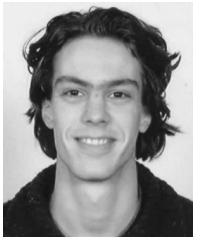
[40] S. Chretien and A.O. Hero, "Kullback Proximal Algorithms for Maximum Likelihood Estimation," *IEEE Trans. Information Theory,* vol. 46, no. 5, pp. 1800-1810, 2000.

[41] R. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation,* vol. 3, pp. 79-87, 1991.

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

[43] B. Kégl, "Intrinsic Dimension Estimation Using Packing Numbers," *Advances in Neural Information Processing Systems,* vol. 15, pp. 681-688, 2003.

[44] J.A. Costa and A.O. Hero, "Geodesic Entropic Graphs for Dimension and Entropy Estimation in Manifold Learning," *IEEE Trans. Signal Processing,* vol. 52, no. 8, pp. 2210-2221, 2004.
[45] E. Levina and P.J. Bickel, "Maximum Likelihood Estimation of Intrinsic Dimension," *Advances in Neural Information Processing Systems,* vol. 17, pp. 777-784, 2005.
[46] J.J. Oliver, R.A. Baxter, and C.S. Wallace, "Unsupervised Learning Using MML," *Proc. Int'l Conf. Machine Learning,* vol. 13, pp. 364-374, 1996.
[47] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 381-396, Mar. 2002.
[48] Z. Ghahramani and M.J. Beal, "Variational Inference for Bayesian Mixtures of Factor Analysers," *Advances in Neural Information Processing Systems,* vol. 12, pp. 449-455, 2000.
[49] J.H. Ham, D.D. Lee, and L.K. Saul, "Semisupervised Alignment of Manifolds," *Proc. Ann. Conf. Uncertainty in Artificial Intelligence,* vol. 10, pp. 120-127, 2005.
[50] H. Ritter, "Parametrized Self-Organizing Maps," *Proc. Int'l Conf. Artificial Neural Networks,* vol. 3, pp. 568-577, 1993.
[51] Z. Ghahramani and M.I. Jordan, "Supervised Learning from Incomplete Data via an EM Approach," *Advances in Neural Information Processing Systems,* vol. 6, pp. 120-127, 1994.
[52] H. Karger, "Riemann Center of Mass and Mollifier Smoothing," *Comm. Pure and Applied Math.,* vol. 3, pp. 509-541, 1977.
[53] A.R. Webb, *Statistical Pattern Recognition.* New York: Wiley, 2002.
[54] G. Peters, B. Zitova, and C. von der Malsburg, "How to Measure the Pose Robustness of Object Views," *Image and Vision Computing,* vol. 20, no. 4, pp. 249-256, 2002.



Jakob Verbeek received a cum laude MSc degree in artificial intelligence (1998) and a cum laude MSc degree in logic (2000), both at the University of Amsterdam, the Netherlands. In 2004, he received a PhD degree from the same university for his research on clustering and dimension reduction. After a postdoctoral researcher position at the University of Amsterdam, he is currently working as a postdoctoral researcher at INRIA Rhone-Alpes. His research interests include latent variable models, their applications in computer vision, and machine learning in general.