Homeomorphic Manifold Analysis (HMA): Generalized Separation of Style and Content on Manifolds

Ahmed Elgammal [a,2,*], Chan-Su Lee[b]

[a]*Department of Computer Science, Rutgers University, 110 Frelinghuysen Rd, Piscataway, NJ, 08854, USA*
[b]*Department of Electronic Engineering, Yeungnam University*
214-1 Dae-dong, Gyeongsan-si, Gyeongbuk-do, 712-749, Republic of Korea

Abstract

The problem of separation of style and content is an essential element of visual perception, and is a fundamental mystery of perception. This problem appears extensively in different computer vision applications. The problem we address in this paper is the separation of style and content when the content lies on a low dimensional nonlinear manifold representing a dynamic object. We show that such a setting appears in many human motion analysis problems. We introduce a framework for learning parameterization of style and content in such settings. Given a set of topologically equivalent manifolds, the Homeomorphic Manifold Analysis (HMA) framework models the variation in their geometries in the space of functions that maps between a topologically-equivalent common representation and each of them. The framework is based on decomposing the style parameters in the space of nonlinear functions that map between a unified embedded representation of the content manifold and style-dependent visual observations. We show the application of the framework in synthesis, recognition, and tracking of certain human motions that follow this setting, such as gait and facial expressions.

Keywords: Style and Content, Manifold Embedding, Kernel Methods, Human Motion Analysis, Gait Analysis, Facial Expression Analysis

[*]Corresponding author
Email addresses: elgammal@cs.rutgers.edu (Ahmed Elgammal), chansu@ynu.ac.kr (Chan-Su Lee)
[1]Tel: 732-445-2001 ext 0021. Fax: 732-445-0537
[2]This work was funded by NSF award IIS-0328991 and NSF CAREER award IIS-0546372

1. Introduction

The problem of separation of style and content is an essential element of visual perception and is a fundamental mystery of perception [1, 2]. For example, we are able to recognize faces and actions under wide variability in the visual stimuli. While the role of manifold representations is still unclear in perception, it is clear that images of the same object lie on a low-dimensional manifold in the visual space defined by the retinal array. On the other hand, neurophysiologists have found that neural population firing is typically a function of a small number of variables, which implies that population activities also lie on low-dimensional manifolds [1].

In this paper we consider the visual manifolds of biological motion. Despite the high dimensionality of the configuration space, many human motions intrinsically lie on low-dimensional manifolds. This is true if we consider the kinematics of the body, as well as the observed motion through image sequences. Let us consider the observed motion. For example, the silhouette (occluding contour) of a human walking or performing a gesture is an example of a dynamic shape, where the shape deforms over time based on the action being performed. These deformations are restricted by the physical body and the temporal constraints posed by the action being performed. Given the spatial and the temporal constraints, these silhouettes, as points in a high-dimensional visual input space, are expected to lie on a low-dimensional manifold. Intuitively, the gait is a one-dimensional manifold that is embedded in a high-dimensional visual space. This was also shown in [3, 4]. Such a manifold can be twisted and even self-intersect in the high-dimensional visual space. Similarly, the appearance of a face performing expressions is an example of a dynamic appearance that lies on a low-dimensional manifold in the visual input space.

Although the intrinsic body configuration manifold might be very low in dimensionality, the resulting visual manifold (in terms of shape and/or appearance) is challenging to model, given the various aspects that affect the appearance. Examples of such aspects include the body type (slim, big, tall etc.) of the person performing the motion, clothing, viewpoint, and illumination. Such variability makes the task of learning a visual manifold very challenging, because we are dealing with data points that lie on multiple manifolds at the same time: body configuration manifold, viewpoint manifold, body shape manifold, illumination manifold, etc.

The main contribution of this is a novel computational framework for learning a decomposable generative model that explicitly factorizes the intrinsic body con-

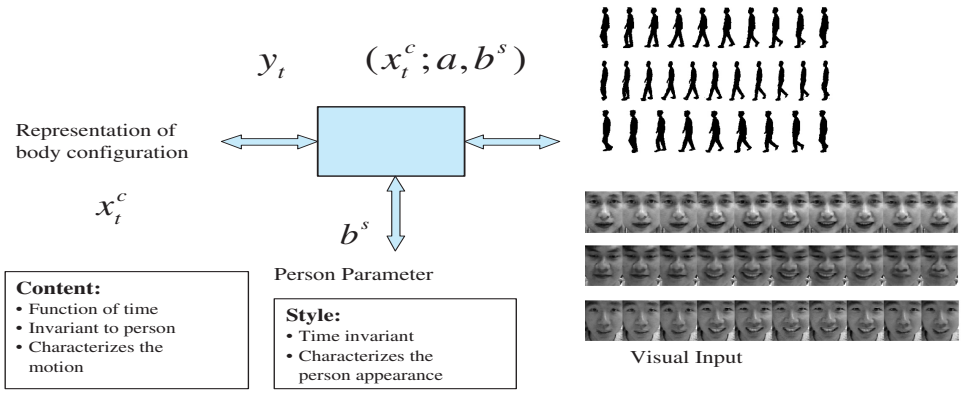


Figure 1: Style and content factors. Content: gait motion or facial expression. Style: different silhouette shapes or face appearance.

figuration (content) as a function of time from the appearance (style) factors. The framework we present in this paper is based on decomposing the style parameters in the space of nonlinear functions that maps between a unified representation of the content manifold and style-dependent observations. Given a set of topologically equivalent manifolds, the Homeomorphic Manifold Analysis (HMA) framework models the variation in their geometries in the space of functions that maps between a topologically-equivalent common representation and each of them. The common representation of the content manifold can be learned from the data or can be enforced in a supervised way if the manifold topology is known. The main assumption here is that the visual manifold is homeomorphic to the unified content manifold representation, and that the mapping between that unified representation and the visual space can be parameterized by different style factors. We describe the motivations and contributions of the framework in more detail within the context of the state-of-the-art in Section 2.

The learned models support tasks such as synthesis and body configuration recovery, as well as the recovery of other aspects such as viewpoint, person parameters, etc. As direct and important applications of the introduced framework, we consider the cases of gait and facial expressions. We show an application of the framework to gait analysis where the model can generate walking silhouettes for different people from different viewpoints. Given a single image or a sequence of images, we can use the model to solve for the body configuration, viewpoint, and person shape style parameters. We also show the application of the framework to facial expressions as an example of a dynamic appearance. In this case, we learn a generative model that generates different dynamic facial expressions for different

people. The model can successfully be used to recognize expressions performed by different people who are not used in model training as well as identifying the person performing the expression.

The paper organization is as follows, Section 2 discusses the relation between the proposed framework and the state-of-the-art. Section 3 summarizes the framework and its applications. Sections 4 describes different ways to obtain unified content manifold representations in both unsupervised and supervised ways. Section 5 describes the details for learning the factorized model. Section 6 describes algorithms for solving for multiple factors. Section 7 shows experimental results and examples of applying the model for dynamic shape and appearance manifolds, for the analysis of gait and facial expressions.

2. Relation to State-of-the-Art

This section puts the contributions of the paper in the context of the state-of-the-art in related areas.

2.1. Factorized Models: Linear, Bilinear, and Multi-linear Models

Linear models, such as PCA [5], have been widely used in appearance modeling to discover subspaces for appearance variations. For example, PCA has been used extensively for face recognition, such as [6–9] and to model the appearance and view manifolds for 3D object recognition, as in [10]. Such subspace analysis can be further extended to decompose multiple orthogonal factors using bilinear models and multilinear tensor analysis [11, 12]. The pioneering work of Tenenbaum and Freeman [11] formulated the separation of style and content using a bilinear model framework [13]. In that work, a bilinear model was used to decompose face appearance into two factors: head pose and different people as style and content interchangeably. They presented a computational framework for model fitting using SVD. Bilinear models have been used earlier in other contexts [13, 14]. A bilinear model is a special case of a more general multilinear model. In [12], multilinear tensor analysis was used to decompose face images into orthogonal factors controlling the appearance of the face including geometry (people), expressions, head pose, and illumination using High Order Singular Value Decomposition (HOSVD) [15]. Tensor representation of image data was used in [16] for video compression, and in [17] for motion analysis and synthesis. N-mode analysis of higher-order tensors was originally proposed and developed in [13, 18, 19] and others. The applications of bilinear and multilinear models to

Figure 2: Twenty sample frames from a walking cycle from a side view. Each row represents half a cycle. Notice the similarity between the two half cycles. The right part shows a plot of the distance between the samples. The two dark lines parallel to the diagonal show the similarity between the two half cycles.

decompose variations into orthogonal factors, as in [11, 12], are mainly for static image ensembles.

The question we address in this paper is how to separate the style and content on a manifold representing a dynamic object. Why don't we just use a bilinear model to decompose the style and content in our case, where certain body poses can be denoted as content and different people as style? The answer is that in the case of dynamic (e.g. articulated) objects, the resulting visual manifold is nonlinear. This can be illustrated by considering the example walking cycle in Fig. 2. In this case, the shape temporally undergoes deformations and self-occlusion, which results a nonlinear manifold. The two shapes in the middle of the two rows correspond to the farthest points in the walking cycle kinematically, which are supposedly the farthest points on the manifold, in terms of the distance along the manifold. In the Euclidean visual input space, these two points are very close to each other, as can be noticed from the distance plot on the right of Fig. 2. Because of such nonlinearity, PCA, bilinear, and multilinear models will not be capable of discovering the underlying manifold and decomposing the orthogonal factors. Linear models will not be able to interpolate intermediate poses and/or intermediate styles.

The framework presented in this paper still utilizes bilinear and multilinear analysis. However, we use such analysis in a different way. The content manifold is explicitly represented using an embedded representation, which can be learned from the data or enforced in a supervised way. Given such representation, the style parameters are factorized in the space of nonlinear mapping functions between a representation of the content manifold and the observations. The main advantage of this approach is that, unlike bilinear and multilinear models [11, 12] that mainly discretize the content space, the content in our case can be treated as a continuous domain.

2.2. Manifold Representations

Embedding manifolds to low-dimensional spaces provides a way to explicitly model such manifolds. Learning motion manifolds can be achieved through linear subspace approximation (PCA), as in [20]. PCA has been widely used in appearance modeling, to discover subspaces for appearance variations, and in modeling view manifolds as in [6–8, 10]. Linear subspace analysis can achieve a linear embedding of the motion manifold in a subspace. However, the dimensionality of the subspace depends on the variations in the data, and not on the intrinsic dimensionality of the manifold.

Nonlinear dimensionality reduction, such as isometric feature mapping (Isomap) [2], Locally linear embedding (LLE) [21], Laplacian eigenmaps [22], Manifold Charting [23], Gaussian Process Latent Variable Models GPLVM [24], and others, can achieve an embedding of a nonlinear manifold through changing the metric from the original space to the embedding space, based on the local structure of the manifold. Spectral methods in particular, such as [2, 21, 22], achieve this embeding through constructing an affinity matrix between the data points, which reflects the local manifold structure. Embedding is then obtained through solving an eigenvalue problem on such matrix. It was shown in [25, 26] that these approaches are all instances of kernel-based learning, in particular kernel principle component analysis KPCA[27]. Several approaches have been proposed to embed new data points, denoted be out of sample embedding, e.g. [28].

Nonlinear dimensionality reduction methods are able to embed image ensembles into low-dimensional spaces, where various orthogonal perceptual aspects can be shown to correspond to certain directions or clusters in the embedding space. In this sense, such methods present an alternative solution to the decomposition problems. However, the application of such approaches is limited to embedding of a single manifold, and it is not clear how to factorize orthogonal factors in the embedding space. As we will show, when multiple manifolds exist in the data (for example, corresponding to different people performing the same activity), such methods tend to capture the intrinsic structure of each manifold separately, without generalizing to capture the inter-manifold aspects. This is because, typically, intra-manifold distances are much smaller than inter-manifold distances. The framework we present in this paper can use nonlinear dimensionality reduction to achieve an embedding of each individual manifold. However, our framework extends such approaches to separate the inter-manifold style parameters. We achieve a factorization of the style parameters in the space of nonlinear mapping functions between the embedded mean manifold, or a other unified representation, and the visual inputs. Another fundamental issue that we address in

this paper is the nonlinearity between a perceptual space and its corresponding high-dimensional observations. Since the manifold structure is not always recoverable from the observation, we introduce the notion of "Conceptual" manifold representation, where we use our knowledge about the manifold topology. The observations are assumed to lie on a nonlinearly deformed version of the conceptual representation of the manifold. Manifold learning in this case is learning such deformation. Unlike traditional unsupervised manifold learning approaches, the conceptual manifold representation is a supervised paradigm.

2.3. Manifold-based Models of Human Motion

Researchers have been trying to exploit the manifold structure as a constraint in tasks such as tracking and activity recognition in an implicit way. Learning data manifolds is typically performed in the visual input space, or through intermediate representations. For example, exemplar-based approaches, such as [29], implicitly model nonlinear manifolds through points (exemplars) along the manifold. Such exemplars are represented in the visual input space. Hidden Markov Models (HMM) provide a probabilistic piecewise-linear approximation of observations. In this sense, the hidden states can follow the manifold and, therefore, HMMs model the observation manifolds in implicit ways, e.g. as in [30] and in [31].

In the last few years, there has been increasing interest in exploiting this fact through using intermediate activity-based manifold representations [4, 31–38]. For example in [4], the visual manifold of human silhouette deformations, due to motion, has been learned explicitly and used for recovering the 3D body configuration from silhouettes in a closed-form. In that work, knowing the motion provided a strong prior to constrain the mapping from the shape space to the 3D body configuration space. In [33] learning the manifold was done on the body configuration space to provide constraints for tracking. In both [4] and [33] learning an embedded manifold representation was decoupled from learning the dynamics, and from learning a regression function between the embedding space and the input space. In [38], coupled learning of the representation and dynamics was achieved using Gaussian Process Dynamic Model (GPDM) [39], in which a nonlinear embedded representation and a nonlinear observation model were fitted through an optimization process. GPDM is a very flexible model since both the state dynamics and the observation model are nonlinear. The problem of simultaneously estimating a latent-state representation coupled with a nonlinear dynamic model was earlier addressed in [40]. Similarly, in [37], models that coupled learning the dynamics with embedding were introduced. It was also shown in [36] that learning motion manifolds provides ways to establish correspondences between

subjects observed from different cameras. In contrast to learning motion manifolds, as in [4, 33, 35], learning the shape manifold, as in [41], provides a way to constrain the recovery of body pose from visual input.

Manifold-based representations of the motion can be learned from kinematic data, or from visual data, e.g., silhouettes. The former is suitable for generative model-based approaches and provides better dynamic-modeling for tracking, e.g., [33, 35]. Learning motion manifolds from visual data, as in [4, 36, 42], provides useful representations for recovery and tracking of body configurations from visual input without the need for explicit body models. The approach introduced in [43] learns a representation for both the visual manifold and the kinematic manifold. Learning a representation of the visual motion manifold can be used in a generative manner as in [4] or as a way to constrain the solution space for discriminative approaches as in [41].

3. Factorized Generative Models

This section summarizes the framework and shows some of its applications in the context of human motion. Our objective is to learn a representation for the shape and/or the appearance of dynamic objects that supports tasks such as synthesis, pose estimation, viewpoint estimation, input reconstruction, and tracking. Such a representation will serve as a factorized generative model for dynamic appearance, where we can think of the image appearance (similar argument for shape) of a dynamic object as instances driven from the model.

To illustrate the point, we start with a single factor model, and then move to the general case. Given a set of image sequences, similar to the ones in Fig. 1, representing a motion (such as gesture, facial expression, or activity) where each sequence is performed by one subject, we aim at learning a generative model that explicitly factorizes the following two factors:

1. Content (body pose): A representation of the intrinsic body configuration through the motion as a function of time invariant to the person, i.e., the content characterizes the motion or the activity.
2. Style (people) : A time-invariant person variable that characterizes the person appearance or shape.

Fig. 1 shows an example of such data, where different people perform the same activity The content in these cases is the gait motion or the smile motion, while the style is a person's shape or face appearance respectively. On the other hand, given an observation of a certain person at a certain body pose, and given the

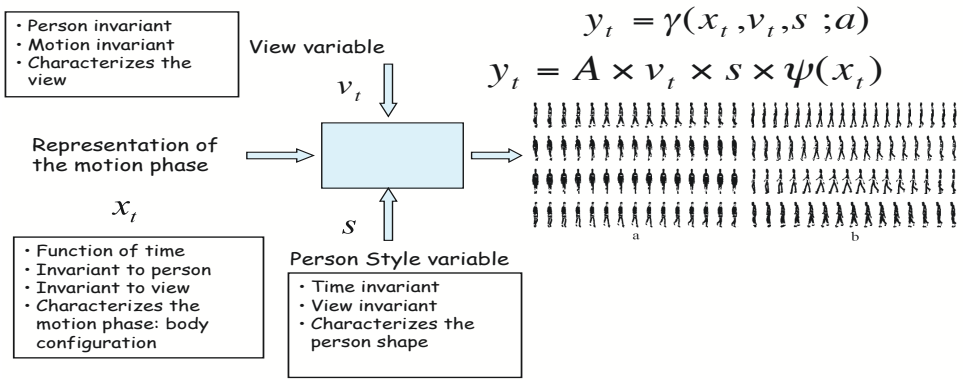


Figure 3: Multiple views and multiple people generative model for gait. a) examples of training data from different views. b) examples of training data for multiple people from the side view.

learned generative model, we aim at solveing for both the body configuration representation (content) and the person shape parameter (style).

In general, the appearance of a dynamic object is a function of the intrinsic body configuration, as well as other factors, such as the object appearance, the viewpoint, illumination, etc. In this paper, we refer to the intrinsic body configuration as the content and all other factors as style factors. The combined appearance manifold, given all these factors, is very challenging to model. Therefore, the solution we use here utilizes the fact that the underline motion manifold, invariant to all other factors, is low in dimensionality. Therefore, the motion manifold can be explicitly modeled, while all the other factors are approximated with subspace models. For example, for the data in Fig. 1, we do not know the dimensionality of the shape manifold of all people, while we know that gait is a one-dimensional manifold motion.

We describe the model for the general case of factorizing multiple style factors given a content manifold. Let $\boldsymbol{y}_t \in \mathbb{R}^d$ be the appearance of the object at time instance t, represented as a point in a d-dimensional space. This instance of the appearance is driven from a generative model in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t, \boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_r; \boldsymbol{a}), \tag{1}$$

where the function $\gamma(\cdot)$ is a mapping function that maps from a representation of body configuration, $\boldsymbol{x}_t$ (content), at time t into the image space given variables $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_r$ each representing a style factor. Such factors are conceptually orthogonal, independent of the body configuration, and can be time variant or invariant. $\boldsymbol{a}$ represents the model parameters.

The data of a particular person's motion at a particular style setting lie on a manifold in the visual space. Let us denote this manifold by $\mathcal{D}^s$. Here, a style setting is a discrete combination of style values. Suppose that we can learn a unified, style-invariant, embedded representation of the motion manifold (content) $\mathcal{M}$ in a low-dimensional Euclidean embedding space, $\mathbb{R}^e$, where such manifold is topologically equivalent, i.e., homeomorphic, to each date manifold $\mathcal{D}^s$. Therefore, each data manifold $\mathcal{D}^s$ is a deformed version of $\mathcal{M}$. We can learn a style-dependent nonlinear mapping functions from $\mathcal{M}$ to each input manifold $\mathcal{D}^s$. Let us denote such mappings by the functions $\gamma_s(\cdot) : \mathbb{R}^e \rightarrow \mathbb{R}^d$ that maps from embedding space of $\mathcal{M}$ into the input space (observation) with dimensionality d for each style setting s. Each of these mapping functions represents a homeomorphism between $\mathcal{M}$ and $\mathcal{D}^s$[3].

In this model, the relation between body configuration and the input is nonlinear. Therefore, the use of nonlinear mapping is essential since the embedding of the configuration manifold is nonlinearly related to the input. Such functions admit a representation in the form of linear combination of basis function [44] and can be written as

$$\boldsymbol{y}_t = \gamma_s(\boldsymbol{x}_t) = \boldsymbol{C}^s \cdot \psi(\boldsymbol{x}_t) \ , \tag{2}$$

where $\boldsymbol{C}^s$ is a $d \times N_\psi$ linear mapping and $\psi(\cdot) : \mathbb{R}^e \rightarrow \mathbb{R}^{N_\psi}$ is a nonlinear kernel map from a representation of the body configuration to a kernel induced space with dimensionality N_ψ. In the mapping in Eq. 2 the style variability is encoded in the coefficient matrix $\boldsymbol{C}^s$. Therefore, the mapping provides a parameterization of all the data manifolds, $\mathcal{D}^s$, in the space of the matrices $\boldsymbol{C}^s$, where each manifold is a point in that space. Given a set of style-dependent functions in the form of Eq. 2, the style variables can be factorized in the linear mapping coefficient space using multilinear analysis of the coefficient tensor. Therefore, the general form for the mapping function $\gamma(\cdot)$ that we use is

$$\gamma(\boldsymbol{x}_t, \boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_r; \boldsymbol{a}) = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{b}_1 \times \cdots \times_r \boldsymbol{b}_r \cdot \psi(\boldsymbol{x}_t) \tag{3}$$

where each $\boldsymbol{b}_i \in \mathbb{R}^{n_i}$ is a vector representing a parameterization of the ith style factor. $\boldsymbol{\mathcal{A}}$ is a core tensor of order $r+2$ and of dimensionality $d \times n_1 \times \cdots \times n_r \times N_\psi$. The product operator $\times_i$ is *mode-i* tensor product as defined in [15].

[3] A function $f : X \rightarrow Y$ between two topological spaces is called a homeomorphism if it is a bijection, continuous, and its inverse is continuous. In our case the existence of the inverse is assumed but not required for computation, i.e., we do not need the inverse for recovering the body configuration or the style parameters. We mainly care about the mapping in a generative manner from $\mathcal{M}$ to $\mathcal{D}^s$.

Since the model is generative (from embedding to visual input) and a nonlinear mapping is used, any representation can be used to model the content manifold as long as it is homeomorphic to the actual manifold. Of course the visual manifold can be degenerate in some cases or it can be self intersecting, because of the projection from 3D to 2D. In such cases the homeomorphic assumption does not hold, however the mapping from $\mathcal{M}$ to $\mathcal{D}^s$ still exists. In such cases the recovery of body configuration and style variables might be ambiguous from a single frame. However, this ambiguity can be overcome if we considered the temporal aspect of the motion.

The model in Eq. 3 can be seen as a hybrid model that uses a mix of nonlinear and multilinear factors. In the model in Eq. 3, the relation between body configuration and the input is nonlinear where other factors are approximated linearly through high-order tensor analysis. The use of nonlinear mapping is essential since the embedding of the configuration manifold is nonlinearly related to the input. The main motivation behind the hybrid model is: The motion itself lies on a low-dimensional manifold, which can be explicitly modeled, while it might not be possible to model the other factors explicitly using nonlinear manifolds. For example, the shapes of different people, although might lie on a manifold; however, we do not know the dimensionality of that manifold and/or we might not have enough data to model it. The best choice is to represent it as a subspace. Therefore, the model in Eq. 3 gives a tool that combines manifold-based models, where manifolds are explicitly embedded, with subspace models for style factors if no better models are available. The framework also allows modeling any style factor on a manifold in its corresponding subspace, since the data can lie naturally on a manifold in that subspace. This feature of the model was further developed in [43], the view manifold of a motion was modeled in the subspace defined by the factorization above.

To achieve the decomposition in Eq 3, we need to learn a unified style-invariant embedded representation of the motion manifold. Several approaches can be used to achieve such a representation, as will be described in Section 4. Nonlinear dimensionality reduction can used to obtain manifold embeddings in an unsupervised manner, and then a mean manifold can be computed as a unified representation through nonlinear warping of the embedded manifold points. Alternatively, supervised conceptual representations can be used as well if the topology of the manifold is known.

For learning the models in this paper, since the goal is to model the manifold of the intrinsic motion due to body configuration changes, we assume there is no large global transformation between images in the training data. All the images

are roughly aligned. The learned models can then be used with actual image data with global transformations, where such transformations can be estimated [45].

In the following, we show some examples of the model in the context of human motion analysis with different roles of the style factors. In the following sections we describe the details for fitting such models and estimation of the parameters. Section 4 describes different ways to obtain a unified nonlinear embedding of the motion manifold for style analysis. Sections 5 describes learning the model. Section 6 describes using the model for solving for multiple factors.

3.1. Example 1: A Single Style Factor Model

Here we give an example of the model with a single style factor. Fig. 1 shows an example of such data, where different people are performing the same activity. The content in this case is the gait motion or the smile motion, while the style is the person shape or face appearance, respectively. The style is a time-invariant variable in this case. The generative model in Eq. 3 reduces to a model in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t^c, \boldsymbol{b}^s; \boldsymbol{a}) = \mathcal{A} \times_2 \boldsymbol{b}^s \times_3 \psi(\boldsymbol{x}_t^c) \ , \tag{4}$$

where the image, $\boldsymbol{y}_t$, at time t is a function of body configuration $\boldsymbol{x}_t^c$ (content) at time t and style variable $\boldsymbol{b}^s$ that is time invariant parameterization of the different motion manifolds. In this case the content is a continuous domain while style is represented by the discrete style classes that exist in the training data. The model parameter is the a third order core tensor, $\mathcal{A}$, with dimensionality $d \times J \times N_\psi$, where J is the dimensionality of the style vector $\boldsymbol{b}^s$, which is the subspace of the different people shapes factored out in the space of the style dependent functions in Eq. 2.

3.2. Example 2: Multifactor Gait Model

As an example of a two-style-factor model, we consider the gait case, with multiple views and multiple people (as shown in Fig. 3). The data set has three components: personalized shape style, viewpoint, and the body configuration. A generative model for silhouettes of different people walking, observed from different viewpoints will be in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t, \boldsymbol{v}_t, \boldsymbol{s}; a) = \mathcal{A} \times \boldsymbol{v}_t \times \boldsymbol{s} \times \psi(\boldsymbol{x}_t), \tag{5}$$

where $\boldsymbol{v}_t$ is a parameterization of the view, which is independent of the body configuration but can change over time, and also independent of the person's shape. $\boldsymbol{s}$ is a time-invariant parameterization of the shape style of the person performing

the walk, independent of the body configuration and the viewpoint. The body configuration $\boldsymbol{x}_t$ evolves along a representation of the gait manifold. In such case the tensor $\mathcal{A}$ is a 4^{th} order tensor with dimensionality $d \times n_v \times n_s \times N_\psi$, where n_v is the dimensionality of the view subspace and n_s is the dimensionality of the shape style subspace.

3.3. Example 3: Multifactor Facial Expressions

Another example is modeling the manifolds of facial expression. Take dynamic facial expressions, such as sad, surprised, happy, etc., where each expression starts from a neutral pose and evolves to a peak expression; each of these motions evolves along a one-dimensional manifold. However, the manifold will be different for each person and for each expression. Therefore, we can use a generative model to generate different people faces and different expressions using a model in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t, \boldsymbol{e}, \boldsymbol{f}; a) = \mathcal{A} \times \boldsymbol{e} \times \boldsymbol{f} \times \psi(\boldsymbol{x}_t) \tag{6}$$

where $\boldsymbol{e}$ is an expression vector (joy, sadness, etc.) that is time-invariant and person-invariant, i.e., it only describes the expression type. Similarly, $\boldsymbol{f}$ is a vector describing a person's facial appearance, which is time-invariant and expression-invariant. The motion content is described by $\boldsymbol{x}_t$ which denotes the motion phase of the expression, i.e., starts from neutral and evolves to a peak expression depending on the expression vector, $\boldsymbol{e}$.

4. Content Manifold Embedding

In order to model the manifold of the body configuration though the motion according to our framework, an embedded representation of that manifold is needed. There are several ways such an embedding can be achieved. The discussion in this section highlights the requirements for that embedding. There are three ways that can be used to achieve such an embedding:

1. *Nonlinear Dimensionality Reduction from Visual Data:* Such techniques assume the data itself, in the observation space, lies on a low-dimensional manifold that is recoverable from the visual data. This might not be always true with the existence of many factors affecting the visual data. This also depends on the representation of the input. This approach, its applicability, and limitations are discussed in Section 4.1.

2. *Supervised Conceptual Embedding:* In many cases the topology of the body configuration manifold is known. While the actual manifold might not be recoverable from the data itself, our conceptual knowledge about the motion manifold allows us to model the data as lying on a distorted or deformed manifold, whose original topology is known. This can be achieved using a conceptual representation of the manifold and using nonlinear mapping to model the deformation of that manifold to fit the data. This approach, its applicability, and limitations are discussed in Section 4.2.
3. *Embedding from Auxiliary Data:* In many cases, both motion capture data as well as visual data are available. The motion capture data (kinematics) can be used to achieve an embedding of the configuration manifold invariant of the aspects affecting the visual observations (viewpoint, style, etc.). The visual data is assumed to be lying on deformed manifolds that are homeomorphic to the configuration manifold. We do not discuss this approach in this paper, we refer the reader to [43] for details.

4.1. Unsupervised Data-driven Manifold Embedding

4.1.1. Nonlinear Dimensionality Reduction

There are a variety of nonlinear dimensionality reduction techniques that can be used to achieve an embedding of the configuration manifold; e.g., LLE [21], Isomap [2], GPLVM [24], etc. All these approaches are unsupervised, where the goal is to achieve a low-dimensional embedded representation of the data, which is presumed to lie on low-dimensional manifolds. Such approaches have been used to achieve embedded representations for tracking and pose estimation, as in [4, 33, 35, 41], etc. These approaches provide a latent variable representation that is nonlinearly related to the data. An important point that we need to emphasize is that the choice of the embedding technique is not fundamental to the approach we introduce here. We mainly assume that an embedding of the data, which preserves local structure of the manifold, can be achieved in a Euclidean space.

Given style-dependent sequences of the same motion under different style setting, an embedding of each sequence can be achieved using nonlinear dimensionality reduction. Since each sequence corresponds to a single style setting (e.g. a certain view and a certain person) the sequence is expected to only show the intrinsic motion manifold. For example, for the case of gait, it was shown in [4, 46] that an embedded representation can be achieved, from visual data, in a three-dimensional Euclidian space using LLE and Isomap. Fig. 4-a shows an example of embedding a walking cycle with 300 frames from a side view. We use a three-dimensional embedding since this is the least-dimensional embedding that can

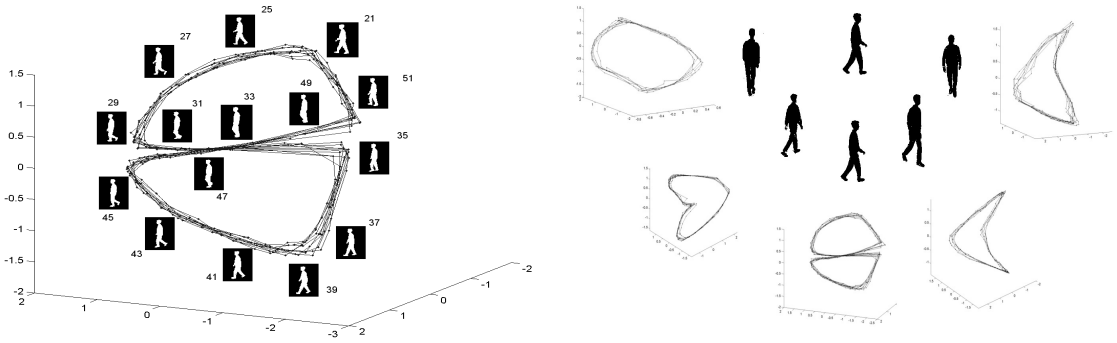


Figure 4: Embedding the Gait Manifold. Left: Embedded gait manifold for a side view of the walker. Sample frames from a walking cycle along the manifold with the frame numbers shown to indicate the order. Ten walking cycles are shown (300 frames). Right: Embedded gait manifolds from five different viewpoints of the walker.

discriminate the different body poses through the cycle. As can be noticed, the embedding can discriminate the two half cycles despite the similarity (e.g., notice that frames 25 and 39 are embedded as the farthest points on the manifold despite the visual similarity between these two instances). One point that need to be emphasized is that we do not use the temporal relation to achieve such an embedding, since the goal is to obtain an embedding that preserves the geometry of the manifold. Temporal relation can be used to determine the neighborhood of each shape but that can lead to erroneous, artificial embedding. This is because it enforces temporally-local neighborhood structure over actual geometric structure (e.g across different cycles). Temporal information can be used to learn dynamics as was shown in [37].

4.1.2. Embedding Multiple Manifolds

Given sequences for different style settings, e.g., different people, different viewpoints, we need to obtain a unified embedding for the underlying body configuration manifold. Nonlinear dimensionality reduction cannot directly obtain a useful embedding with multiple style settings existing in the data (because such data itself will not exhibit the manifold structure that we expect to capture). Nonlinear dimensionality reduction techniques cannot directly embed multiple people's manifolds simultaneously in a way that yield a useful representation. Although such approaches try to capture the manifold geometry, typically, the intra-manifold distances are much smaller compared to the inter-manifold distances. An example is shown in Fig. 5-a where LLE is used to embed three people's man-

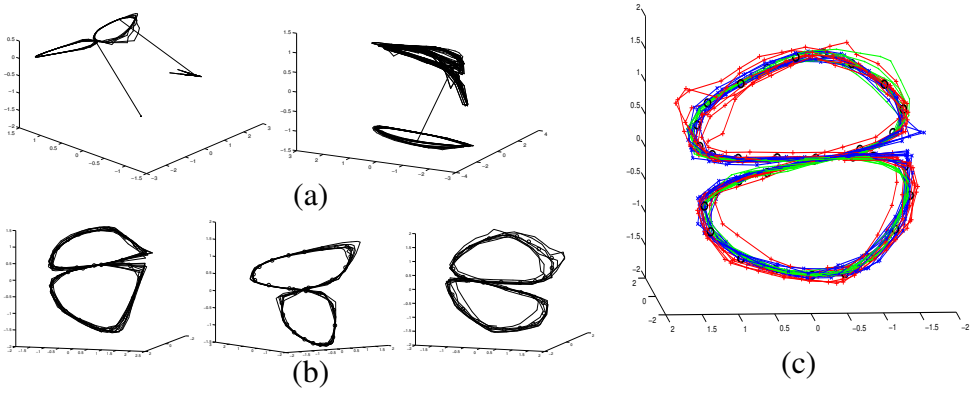


Figure 5: a) Embedding obtained by LLE for three people data with two different K values. Inter-manifold distance dominates the embedding. b) Separate embedding of three manifolds for three people data. c) Unified manifold embedding $\tilde{X}^k$

ifolds where all the inputs are spatially registered. As a result, the embedding shows separate manifolds (e.g., in the left figure one manifold is degenerate to a point because the embedding is dominated by the manifold with largest intra-manifold distance.) Even if we force LLE to include corresponding points on different manifolds to each point's neighbors, this results in superficial embedding that does not capture the manifold geometry. This is an instance of the problem know as manifold alignment. Another fundamental problem is that different people will have different manifolds because their appearance (shape) is different, which imposes different twists to the manifolds and, therefore, different geometry. This can be noticed in Fig. 11-b.

To achieve a unified embedding of a certain activity manifold from multiple people data, each person's manifold is embedded separately using LLE. Each manifold is represented as a parametric curve. Given the embedded manifold $\boldsymbol{X}^k$ for person k, a cubic spline $\boldsymbol{m}^k(t)$ is fitted to the manifold as a function of time, i.e., $\boldsymbol{m}^k(t) : \mathbb{R} \rightarrow \mathbb{R}^e$, where $t = 0 \rightarrow 1$ is a time variable. The manifold for person k is sampled at N uniform time instances $\boldsymbol{m}^k(t_i)$, where $i = 1 \cdots N$. For the case of periodic motion, such as gait, each cycle on the manifold is time mapped from 0 to 1 given a corresponding origin point on the manifold, where the cycles can be computed from the geodesic distances to the origin.

Given multiple manifolds, a mean manifold $Z(t_i)$ is learned by warping $\boldsymbol{m}^k(t_i)$ using non-rigid transformation using an approach similar to [47]. We solve for a mean manifold $Z(t_i)$ and a set of regularized non-rigid transformations $f(.;\alpha_k)$

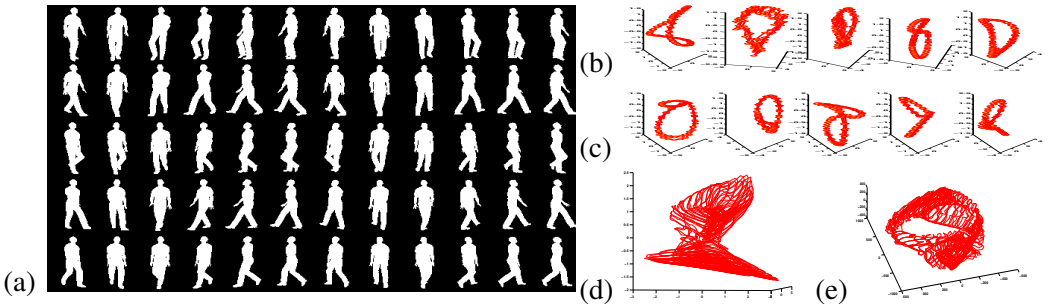


Figure 6: Data-driven view and body configuration manifolds:(a) Examples of sample data with view and configuration variations. Rows: body pose at $0, \frac{1}{5}T, \frac{2}{5}T, \frac{3}{5}T, \frac{4}{5}T$, where T is a walking cycle period. Cols:view $0, 30, 60, \cdots, 330$. (b) Intrinsic configuration manifold when view angle is $0, 60, 120, 180$, and240. (c) View manifold for five different fixed body pose. (d) (e) Combined view and body configuration manifold by LLE and Isomap.

where the objective is to minimize the energy function

$$E(f) = \sum_k \sum_i \|Z(t_i) - f(\boldsymbol{m}^k(t_i); \alpha_k)\|^2 + \lambda \|Lf\|^2 \tag{7}$$

where λ is a regularization parameter and $\|Lf\|^2$ is a smoothness term. In particular thin-plate spline (TPS) is used for the nonrigid transformation. Given the transformation parameters α_k, the whole data sets are warped to obtain a unified embedding $\tilde{\boldsymbol{X}}^k$ for the k manifolds, where $\tilde{\boldsymbol{X}}^k = f(\boldsymbol{X}^k; \alpha_k), k = 1 \cdots K$. Fig. 5-b,c shows an example of three different manifolds and their warping into a unified manifold embedding. In general, we found that this warping solution is suitable for a single factor model. In multiple factor models the deformations can be quite large among the multiple manifolds representing the different variant factors. In such case, a conceptual embedding is preferred. Alternative solution for embedding multiple manifolds by modifying spectral nonlinear dimensionality reduction techniques to capture both the inter- and intra- manifold geometry was proposed in [48].

4.2. Supervised Conceptual Manifold Embedding

One essential limitation of using nonlinear dimensionality reduction to achieve an embedding of the visual manifold is that the data itself might not lie on a manifold in the visual space, as we think it should, due to different reasons including noise, image representations, existence of other sources of variability that are not counted for, etc. In contrast to unsupervised learning of the content manifold representation described above, if the topology of the manifold is known, a

conceptual topologically-equivalent representation of the manifold can be directly used. By topologically-equivalent, we mean equivalent to our notion of the underlying motion manifold. The actual data is a deformed version of that manifold, where such deformation is captured through the nonlinear mapping in Eq. 2 in a generative way. Here, we explain our motivation behind this approach.

The general model in Eq. 3 requires a unified representation of the content manifold. However, since the visual manifolds twist very differently given each factor (different people or different views, etc.), it is not possible to achieve a unified configuration manifold representation independent of other factors. This is shown in Fig. 6, which shows examples of embedding gait manifold of different views and embedding of both the gait and view manifold. These limitations motivate the use of a conceptual unified representation of the configuration manifold that is invariant to all other factors. This unified representation would allow the model in Eq. 3 to generalize to as many factors as desired.

For example for the gait case in Eq. 5, the body configuration x_t evolves along a representation of the manifold, which is supposed to be homeomorphic to the actual gait manifold. The question is, what conceptual representation of the manifold can we use? Since the gait is a one-dimensional closed manifold embedded in the input space, we can think of it as a unit circle twisted and stretched in the space based on the shape and the appearance of the person under consideration, or based on the viewpoint. In general, all closed 1D manifolds are topologically homeomorphic to a unit circle. So, we can use a unit circle as a unified representation of all gait cycles for all people for all views. Given that all the manifolds under consideration are homeomorphic to the unit circle, the actual data is used to learn nonlinear warping between the conceptual representation and the actual data manifold.

One important thing to notice is since the mapping in Eq. 2 is from the representation to the data, it will always be a function. Therefore, even if the manifold in the observation space might have a different topology, e.g. self-intersecting or collapsing, this will not be a problem in learning the manifold deformation. Such conceptual representation of the manifold was successfully used to model image translations and rotations in [49]. Other topologies can also be used to model more complex manifolds [50].

5. Generalized Style factorization

5.1. Style Setting

To fit the model in Eq. 3 we need image sequences at each combination of style factors, all representing the same motion (content). The input sequences do not have to be of the same length. Each style factor is represented by a set of discrete samples in the training data, i.e., a set of discrete views, discrete shape styles, discrete expressions, etc. We denote the set of discrete samples for the ith style factor by B_i and the number of these samples by $N_i = |B_i|$. A certain combination of style factors is denoted by an r-tuple $s \in B_1 \times \cdots \times B_r$. We call such a tuple "*a style setting*." Overall, the training data needed to fit the model is of size $N_1 \times \cdots \times N_r$ sequences.

5.2. Learning Style Dependent Mappings

Let the sets of input image sequences be $\mathsf{Y}^s = \{\boldsymbol{y}_i^s \in \mathbb{R}^d \quad i = 1, \cdots, n_s\}$ where s is the style setting index(as defined above), n_s is the length of the sequence, and d is the input dimensionality. Let the corresponding points on the unified embedding space be $\mathsf{X}^s = \{\boldsymbol{x}_i^s \in \mathbb{R}^e, \quad i = 1, \cdots, n_s\}$, where e is the dimensionality of the embedding space.

We consider the case for sth sequence. We will drop the index s when it is implied from the context for simplicity. Given a style-specific sequence Y^s and its embedding coordinates X^s, we learn a style-dependent nonlinear mapping function from the embedding space into the input space, i.e., a function $\gamma_s(\cdot) : \mathbb{R}^e \rightarrow \mathbb{R}^d$ that maps from embedding space into the input space (observation). We can learn a nonlinear mapping function $\gamma_s(\cdot)$ that satisfies $\boldsymbol{y}_i^s = \gamma_s(\boldsymbol{x}_i^s), i = 1 \cdots n_s$ and minimizes a regularized risk criteria. From the representer theorem [44], such a function admits a representation of the form of linear combination of basis functions around arbitrary points $\boldsymbol{z}_j \in \mathbb{R}^e$, $j = 1 \cdots N$ on the manifold. In particular we use a semi-parametric form for the function $\gamma(\cdot)$. Therefore, for the l-th dimension of the input (l-th pixel), the function $\gamma^l(\cdot)$ is an RBF interpolant from $\mathbb{R}^e$ into $\mathbb{R}$ in the form

$$\gamma^l(\boldsymbol{x}) = p^l(\boldsymbol{x}) + \sum_{i=1}^{N} w_j^l \phi(|\boldsymbol{x} - \boldsymbol{z}_j|), \tag{8}$$

where $\phi(\cdot)$ is a real-valued basic function, w_j are real coefficients, $|\cdot|$ is the second norm on $\mathbb{R}^e$ (the embedding space). The choice of the centers is arbitrary (not necessarily data points). Therefore, This is a form of Generalized Radial Basis Function (GRBF) [51].

Typical choices for the basis (kernel) function include thin-plate spline ($\phi(u) = u^2 log(u)$), the multiquadric ($\phi(u) = \sqrt{u^2 + a^2}$), Gaussian ($\phi(u) = e^{-au^2}$), biharmonic ($\phi(u) = u$) and triharmonic ($\phi(u) = u^3$) splines[4]. p^l is a linear polynomial with coefficients c^l, i.e., $p^l(x) = [1 \quad \boldsymbol{x}^\top] \cdot c^l$.

The whole mapping can be written in a matrix form as

$$\gamma_s(\boldsymbol{x}) = \boldsymbol{C}^s \cdot \psi(\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{C}^s$ is a $d \times (N+e+1)$ dimensional matrix with the l-th row $[w_1^l \cdots w_N^l \quad c^{l^\top}]$. The vector $\psi(\boldsymbol{x})$ represents a nonlinear kernel map from from the embedded representation of the body configuration (content manifold) to a kernel induced space, i.e., from $\mathbb{R}^e$ to $\mathbb{R}^{N_\psi}$. The kernel map $\psi()$ is defined by the points z_j as

$$\psi(\boldsymbol{x}) = [\phi(|\boldsymbol{x} - \boldsymbol{z}_1|) \cdots \phi(|\boldsymbol{x} - \boldsymbol{z}_N|) \quad 1 \quad \boldsymbol{x}^\top]^\top. \tag{10}$$

In this case the dimensionality of induced kernel space is $N_\psi = N + e + 1$. The matrix $\boldsymbol{C}^s$ represents the coefficients for d different nonlinear mappings for style setting s, each from a low-dimension embedding space into real numbers.

To insure orthogonality and to make the problem well posed, the following side condition constraints are imposed: $\sum_{i=1}^N w_i p_j(\boldsymbol{x}_i) = 0, j = 1, \cdots, m$ where p_j are the linear basis of p. Therefore the solution for $\boldsymbol{C}^s$ can be obtained by directly solving the linear systems

$$\begin{pmatrix} \boldsymbol{A} + \lambda \boldsymbol{I} & \boldsymbol{P}_x \\ \boldsymbol{P}_t^\top & 0_{(e+1)\times(e+1)} \end{pmatrix}_s \boldsymbol{C}^{s\top} = \begin{pmatrix} \boldsymbol{Y}_s \\ 0_{(e+1)\times d} \end{pmatrix}, \tag{11}$$

where $\boldsymbol{A}, \boldsymbol{P}_x, \boldsymbol{P}_t$ are defined for the s-th style setting as: $\boldsymbol{A}$ is $n_s \times N$ matrix with $\boldsymbol{A}_{ij} = \phi(|\boldsymbol{x}_i^s - \boldsymbol{z}_j|), \quad i = 1 \cdots n_s, j = 1 \cdots N$, $\boldsymbol{P}_x$ is a $n_s \times (e+1)$ matrix with i-th row $[1 \quad \boldsymbol{x}_i^{s\top}]$, $\boldsymbol{P}_t$ is a $N \times (e+1)$ matrix with i-th row $[1 \quad \boldsymbol{z}_i^\top]$. $\boldsymbol{Y}_s$ is $(n_s \times d)$ matrix containing the input images for style setting s, i.e., $\boldsymbol{Y}_s = [\boldsymbol{y}_1^s \cdots \boldsymbol{y}_{n_s}^s]^\top$. Solution for $\boldsymbol{C}^s$ is guaranteed under certain conditions on the basic functions used.

5.3. Style Factorization

Given the learned nonlinear mapping coefficients $\boldsymbol{C}^s$ for all style settings $s \in B_1 \times \cdots \times B_r$, the style parameters can be factorized by fitting a multilinear

[4]The polynomial part is needed for positive semi-definite kernels to span the null space in the corresponding RKHS. The polynomial part is essential regularizer with the choice of specific basis functions such as Thin-plate spline kernel. A Gaussian kernel does not need a polynomial part. [52]

model [12, 15] to the coefficients' tensor. Higher-order tensor analysis decomposes multiple orthogonal factors, as an extension of principal component analysis (PCA) (one factor), and bilinear model (two orthogonal factors). Singular value decomposition (SVD) can be used for PCA analysis and iterative SVD with *vector transpose* for bilinear analysis [11]. Higher-order tensor analysis can be achieved by higher-order singular value decomposition (HOSVD) with *matrix unfolding*, which is a generalization of SVD [15][5]

Each of the coefficient matrices $\boldsymbol{C}^s$, with dimensionality $d \times N_\psi$ can be represented as a coefficient vector $\boldsymbol{c}^s$ by column stacking, i.e., c^s is an $N_c = d \cdot N_\psi$ dimensional vector. All the coefficient vectors can then be arranged in an order $r+1$ coefficient tensor C with dimensionality $N_c \times N_1 \times \cdots \times N_r$. The coefficient tensor is then factorized using HOSVD as

$$\boldsymbol{C} = \tilde{\mathcal{D}} \times_1 \tilde{\boldsymbol{B}}_1 \times_2 \tilde{\boldsymbol{B}}_2 \times \cdots \times_r \tilde{\boldsymbol{B}}_r \times_{r+1} \tilde{\boldsymbol{F}} \ ,$$

where $\tilde{\boldsymbol{B}}_i$ is the mode-i basis of $\boldsymbol{C}$, which represents the orthogonal basis for the space for the i-th style factor. $\tilde{\boldsymbol{F}}$ represents the basis for the mapping coefficient space. The dimensionality of each of the $\tilde{\boldsymbol{B}}_i$ matrices is $N_i \times N_i$. The dimensionality of the matrix $\tilde{\boldsymbol{F}}$ is $N_c \times N_c$. $\mathcal{D}$ is a core tensor, with dimensionality $N_1 \times \cdots \times N_r \times N_c$, which governs the interactions (the correlation) among the different mode basis matrices.

Similar to PCA, it is desired to reduce the dimensionality for each of the orthogonal spaces to retain a subspace representation. This can be achieved by applying higher-order orthogonal iteration for dimensionality reduction [53]. The reduced subspace representation is

$$\boldsymbol{C} = \mathcal{D} \times_1 \boldsymbol{B}_1 \times \cdots \times_r \boldsymbol{B}_r \times_{r+1} \boldsymbol{F} \ , \tag{12}$$

where the reduced dimensionality for $\mathcal{D}$ is $n_1 \times \cdots \times n_r \times n_c$, for $\boldsymbol{B}_i$ is $N_i \times n_i$, and for $\boldsymbol{F}$ is $N_c \times n_c$, where $n_1, \cdots, n_r$, and n_c are the number of basis retained for each factor respectively. Since the basis for the mapping coefficients, $\boldsymbol{F}$ is not used in the analysis, we can combine it with the core tensor using tensor multiplication to obtain coefficient eigenmodes, which is a new core tensor formed by $\mathcal{Z} = \mathcal{D} \times_{r+1} \boldsymbol{F}$

[5] Matrix unfolding is an operation to reshape high order tensor array into matrix form. Given an r-order tensor $\mathcal{A}$ with dimensions $N_1 \times N_2 \times \cdots \times N_r$, the mode-$n$ matrix unfolding, denoted by $\boldsymbol{A}_{(n)} = unfolding(\mathcal{A}, n)$, is flattening $\mathcal{A}$ into a matrix whose column vectors are the mode-n vectors [15]. Therefore, the dimension of the unfolded matrix $\boldsymbol{A}_{(n)}$ is $N_n \times (N_1 \times N_2 \times \cdots N_{n-1} \times N_{n+1} \times \cdots N_r)$.

with dimensionality $n_1 \times \cdots \times n_r \times N_c$. Therefore, Eq. 12 can be rewritten as

$$\boldsymbol{C} = \mathcal{Z} \times_1 \boldsymbol{B}_1 \times \cdots \times_r \boldsymbol{B}_r \ . \tag{13}$$

The columns of the matrices $\boldsymbol{B}_1, \cdots, \boldsymbol{B}_r$ represent orthogonal basis for each style factor's subspace respectively. Any style setting s can be represented by a set of style vectors $\boldsymbol{b}_1 \in \mathbb{R}^{n_1}, \cdots, \boldsymbol{b}_r \in \mathbb{R}^{n_r}$ for each of the style factors. The corresponding coefficient matrix $\boldsymbol{C}$ can then be generated by unstacking the vector $\boldsymbol{c}$ obtained by the tensor product

$$\boldsymbol{c} = \mathcal{Z} \times_1 \boldsymbol{b}_1 \times \cdots \times_r \boldsymbol{b}_r.$$

Therefore, we can generate any specific instant of the motion by specifying the body configuration parameter $\boldsymbol{x}_t$ through the kernel map defined in Eq. 10. The whole model for generating image $\boldsymbol{y}_t^s$ can be expressed as

$$\boldsymbol{y}_t^s = \mathit{unstacking}(\mathcal{Z} \times_1 \boldsymbol{b}_1 \times \cdot \times_r \boldsymbol{b}_r) \cdots \psi(\boldsymbol{x}_t)$$

This can be expressed abstractly also by arranging the tensor $\mathcal{Z}$ into a order $r+2$ tensor $\mathcal{A}$ with dimensionality $d \times n_1 \times \cdots \times n_r \times N_\psi$. The results in the factorization in the form of Eq. 3, i.e.,

$$\boldsymbol{y}_t^s = \mathcal{A} \times_1 \boldsymbol{b}_1 \times \cdots \times_r \boldsymbol{b}_r \cdots \psi(\boldsymbol{x}_t) \ .$$

6. Solving for Multiple Factors

Given a multi-factor model fitted as described in the previous section, and given a new image or a sequence of images, it is desired to efficiently solve for each of the style factors, as well as the body configuration. We discriminate here between two cases: 1) *The input is a whole motion cycle,* 2) *The input is a single image.* For the first case, since we have a whole motion manifold, we can obtain a closed-form analytical solution for each of the factors by aligning the input sequence manifold to the model manifold representation. For the second case, we introduce an iterative deterministic annealing solution. Alternatively, sampling methods such as MCMC and Particle Filter can be used for inferring the body configuration and style parameters from a single image, or through a temporal sequence of frames, e.g. [43, 50].

6.1. Solving for Style Factors Given a Whole Sequence

Given a sequence of images, representing a whole motion cycle, we can solve for the different style factors iteratively. First the sequence is embedded and aligned to the embedded content manifold. Then, the mapping coefficient matrix $\boldsymbol{C}$ is learned from the aligned embedding to the input. Given such coefficients, we need to find the optimal $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_r$ factors, which can generate such coefficients, i.e., minimizes the error

$$E(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_r) = \|\boldsymbol{c} - \mathcal{Z} \times_1 \boldsymbol{b}_1 \times_2 \cdots \times_r \boldsymbol{b}_r\| \tag{14}$$

where $\boldsymbol{c}$ is the column stacking of $\boldsymbol{C}$. If all the style vectors are known except the ith factor's vector, then we can obtain a closed-form solution for $\boldsymbol{b}_i$. This can be achieved by evaluating the product

$$\mathcal{G} = \mathcal{Z} \times_1 \boldsymbol{b}_1 \times \cdots \times_{i-1} \boldsymbol{b}_{i-1} \times_{i+1} \boldsymbol{b}_{i+1} \times \cdots \times_r \boldsymbol{b}_r$$

to obtain a tensor $\mathcal{G}$. Solution for $\boldsymbol{b}_i$ can be obtained by solving the system $\boldsymbol{c} = \mathcal{G} \times_2 \boldsymbol{b}_i$ for $\boldsymbol{b}_i$, which can be written as a typical linear system by unfolding $\mathcal{G}$ as a matrix. Therefore, estimate of $\boldsymbol{b}_i$ can be obtained by

$$\boldsymbol{b}_i = (\boldsymbol{G}_2)^{\dagger} \boldsymbol{c} \tag{15}$$

where $\boldsymbol{G}_2$ is the matrix obtained by mode-2 unfolding of $\mathcal{G}$ and $\dagger$ denotes the pseudo-inverse using SVD. Similarly, we can analytically solve for all other style factors. We start with a mean style estimate for each of the style factors, since the style vectors are not known at the beginning. Iterative estimation of each of the style factors using Eq. 15 would lead to a local minima for the error in Eq. 14.

6.2. Solving for Body Configuration and Style Factors from a Single Image

In this case the input is a single image $\boldsymbol{y} \in \mathbb{R}^d$, and it is required to find the body configuration, (i.e., the corresponding embedding coordinates $\boldsymbol{x} \in \mathbb{R}^e$ on the manifold) and the style factors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_r$. These parameters should minimize the reconstruction error defined as

$$E(\boldsymbol{x}, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_r) = \|\boldsymbol{y} - \mathcal{A} \times_1 \boldsymbol{b}_1 \times \cdots \times_r \boldsymbol{b}_r \times_{r+1} \psi(\boldsymbol{x})\|^2 \tag{16}$$

Instead of the second norm, we can also use a robust error metric and, in both cases, we end up with a nonlinear optimization problem.

One challenge is that not every point in a style subspace is a valid style vector. For example, if we consider a shape stye factor, we do not have enough data to

model the manifold of all human shapes in this space. Training data, typically, is just a very sparse sampling of this manifold. To overcome this, we assume, for all style factors, that the optimal style can be written as a convex linear combination of the style classes in the training data. This assumption is necessary to constrain the solution space. Better constraints can be achieved with sufficient training data. For example, we can model the viewpoint manifold in the view factor subspace given sufficient sampled viewpoints.

For the i-th style factor, let the mean vectors of the style classes in the training data denoted be $\bar{\boldsymbol{b}}_i^k, k = 1, \cdots, K_i$, where K_i is the number of classes and k is the class index. Such classes can be obtained by clustering the style vectors for each style factor in its subspace. Given such classes, we need to solve for linear regression weights α_{ik} such that

$$\boldsymbol{b}_i = \sum_{k=1}^{K_i} \alpha_{ik} \bar{\boldsymbol{b}}_i^k \ .$$

If all the style factors are known, then Eq. 16 reduces to a one-dimensional search problem for the body configuration $\boldsymbol{x}$ on the embedded manifold representation that minimizes the error. On the other hand, if the body configuration and all style factors are known except the i-th factor, we can obtain the conditional class probabilities $p(k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i})$, which is proportional to observation likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i}, k)$. Here, we use the notation $\boldsymbol{s}_{/\boldsymbol{b}_i}$ to denote the style factors excluding the i-th factor. This likelihood can be estimated assuming a Gaussian density centered around $\mathcal{A} \times_1 \boldsymbol{b}_1 \times \cdots \times_i \bar{\boldsymbol{b}}_i^k \times \cdots \times_r \boldsymbol{b}_r \times \psi(\boldsymbol{x})$ with covariance Σ_{ik}, i.e.,

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i}, k) \approx \mathcal{N}(\mathcal{A} \times_1 \boldsymbol{b}_1 \times \cdots \times_i \boldsymbol{b}_i^k \times \cdots \times_r \boldsymbol{b}_r \times \psi(\boldsymbol{x}), \Sigma_{ik}).$$

Given view class probabilities, the weights are set to $\alpha_{ik} = p(k \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i})$. This setting favors an iterative procedure for solving for $\boldsymbol{x}, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_r$. However, wrong estimation of any of the factors would lead to wrong estimation of the others, then leading to a local minima. For example, in the gait model in section 3.2, a wrong estimation of the view factor would lead to a totally wrong estimate of body configuration, and a wrong estimate for shape style. To avoid this we use a deterministic annealing-like procedure, where at the beginning the weights for all the style factors are forced to be close to uniform to avoid hard decisions. The weights gradually become discriminative thereafter. To achieve this, we use variable class variances, which are uniform to all classes and are defined as $\Sigma_i = T\sigma_i^2 \boldsymbol{I}$ for the i-th factor. The temperature parameter T starts with a large value and gradually reduced in each step and a new body configuration estimate is computed. We summarize the solution framework in Fig. 7.

Figure 7: Iterative Estimation of Style Factors

Input: image $\boldsymbol{y}$, style classes' means $\bar{\boldsymbol{b}}_i^k$, for all style factors $i = 1, cdots, r$, core tensor $\mathcal{A}$

Initialization:
- initialize T
- initialize α_{ik} to uniform weights, i.e., $\alpha_{ik} = 1/K_i, \forall i, k$
- Compute initial $\boldsymbol{b}_i = \sum_{k=1}^{K_i} \alpha_{ik} \bar{\boldsymbol{b}}_i^k, \forall i$

Iterate:
- Compute coefficient $\boldsymbol{C} = \mathcal{A} \times \boldsymbol{b}_1 \times \cdots \times \boldsymbol{b}_r$
- Estimate body configuration: 1-D search for $\boldsymbol{x}$ that minimizes $E(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{C}\psi(\boldsymbol{x})\|$
- For style factor $i = 1, \cdots, r$, estimate a new style factor vector $\boldsymbol{b}_i$
  - $\forall k = 1, \cdots K_i$ Compute $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i}, k)$
  - $\forall k$ Update the weights $\alpha_{ik} = p(k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s}_{/\boldsymbol{b}_i})$
  - Estimate new factor vector as $\boldsymbol{b}_i = \sum_{k=1}^{K_i} \alpha_{ik} \boldsymbol{b}_i^k$
- Reduce T

---

7. Applications and Results

In this section we illustrate several examples of the proposed model in different settings. In parallel to section 3, we describe results for the three examples introduced earlier for 1) a single style factor model for gait, 2) a multifactor model for gait, 3) a multifactor model for facial expressions.

For all the experiments reported here on gait, we used CMU Mobo gait data set [54], which contains walking people from multiple synchronized views. The CMU Mobo gait data set contains 25 people (about 8 to 11 walking cycles) captured from six different viewpoints. Each subject walks on a treadmill to capture gait sequences with consistent views using fixed cameras. The shape is represented using implicit function representations as mentioned earlier. All silhouettes used for training are extracted using background subtraction. We also evaluated the performance of different variants of the model using the HumanEva dataset [55], which is a benchmark for quantitative evaluation of pose estimation algorithms. For the experiments reported on facial expressions, we used the CMU-Cohen-Kanada AU facial expression dataset [56].

7.1. Representation:

One essential challenge when modeling visual data manifolds is the issue of image representation. While in principal the data is expected to lie on a low-dimensional manifold, the actual image representation might not exhibit that. The manifold might not be recoverable from the data, if the representation does not exhibit smooth transitions between images that are supposed to be neighboring points on the manifold. In this paper we are dealing with two types of image representations: shapes and appearances. Here, we describe the used representations.

Shape Representation: We represent each shape instance as an implicit function $y(x)$ at each pixel x such that $y(x) = 0$ on the contour, $y(x) > 0$ inside the contour, and $y(x) < 0$ outside the contour. We use a signed-distance function for this purpose. Such a representation imposes smoothness on the distance between shapes. Given such a representation, an input shape is a point in $\mathbb{R}^d$, where d is the dimensionality of the input space. Implicit function representation is typically used in level-set methods.

Appearance Representation: Appearance is represented directly in a vector form of raw pixel intensities, i.e., each instance of appearance is represented as point in R^d where d is the dimensionality of the input space.

7.2. A Single Factor Model for Gait

In this section we describe several experiments for the single style-factor model, as described in section 3.1, and as illustrated in Fig. 1, for gait where the content is the motion and the style is the person shape.

7.2.1. Experiment 1 - Style Dependent Shape Interpolation

The point of this experiment is to show that the model in Eq. 4 can be used to interpolate new shapes at intermediate body configurations with different people shape style, even with a very small number of training samples. We fitted the model in Eq. 4 using three people's silhouettes during a half walking cycle to separate the style (person's shape) from the content (body pose). The input is three sequences containing only 10, 11, and 9 frames respectively. The input silhouettes are shown in Fig. 8-a. Note that the three sequences are not of equal length and the body poses are not necessarily in correspondence. Since the input size in this case is too small to be able to discover the manifold geometry, we embed the data points on a unit circle as a topologically homeomorphic manifold (as an approximation of the manifold of half a cycle) where each sequence is equally spaced along the circle. Embedding is shown in Fig. 8-b. We selected 8 RBF centers at 8 quadrics on the circle. The model is then fitted to the data in

the form of Eq. 4 using TPS kernels. Fig. 8-d shows the RBF coefficients for the three people (one in each row), where the last three columns are the polynomial coefficients. Fig. 8-c shows the style coefficients for the three people and Fig. 8-e shows the content bases.

Given the fitted model we can show some interesting results. First, we can interpolate intermediate silhouettes for each of the three people's styles. This is shown in Fig. 9, where 16 intermediate poses were rendered. Notice that the input contained only 9 to 11 data points for each person. A closer look at the rendered silhouettes shows that model can really interpolate intermediate silhouettes that were never seen as inputs (e.g., person 1 column 4 and person 3 columns 5, 15). We can also interpolate half walking cycles in new styles. This is shown in Fig. 9 where intermediate styles intermediate contents were used.

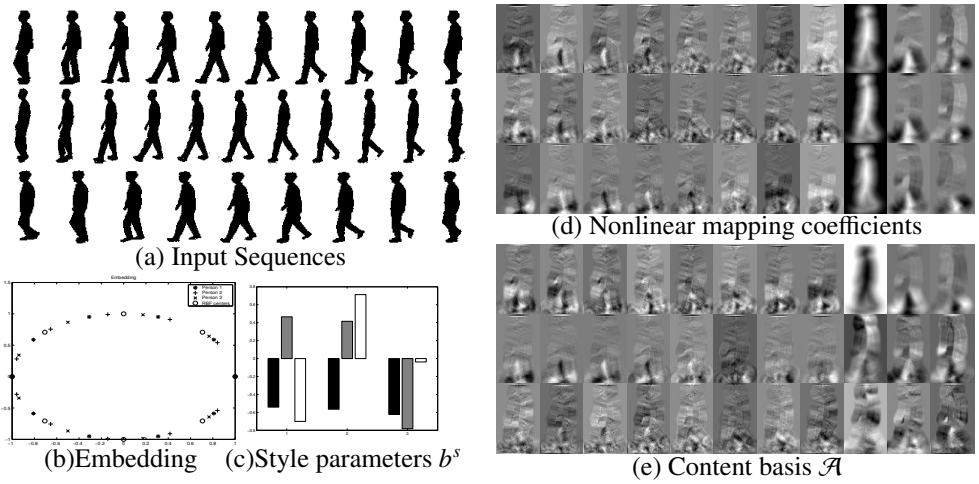


Figure 8: Learning shape style and motion content for a gait example

7.2.2. Experiment 2- Style-Preserving Pose-Preserving Reconstruction

We can use the learned model to reconstruct noisy and corrupted input instances in a way that preserves both the body pose and the person style. Given an input silhouette, we solve for both the embedding coordinate and the style, and then use the model to reconstruct a corrected silhouette given the recovered pose and person parameters. Fig. 9 shows such reconstruction, where we used 48 noisy input silhouettes from CMU Mobogait database (16 for each person shown at each row). The noise in the silhouettes is typical fragmentation and holes resulting from background subtraction. The resulting people's probabilities are shown in Fig. 9-c

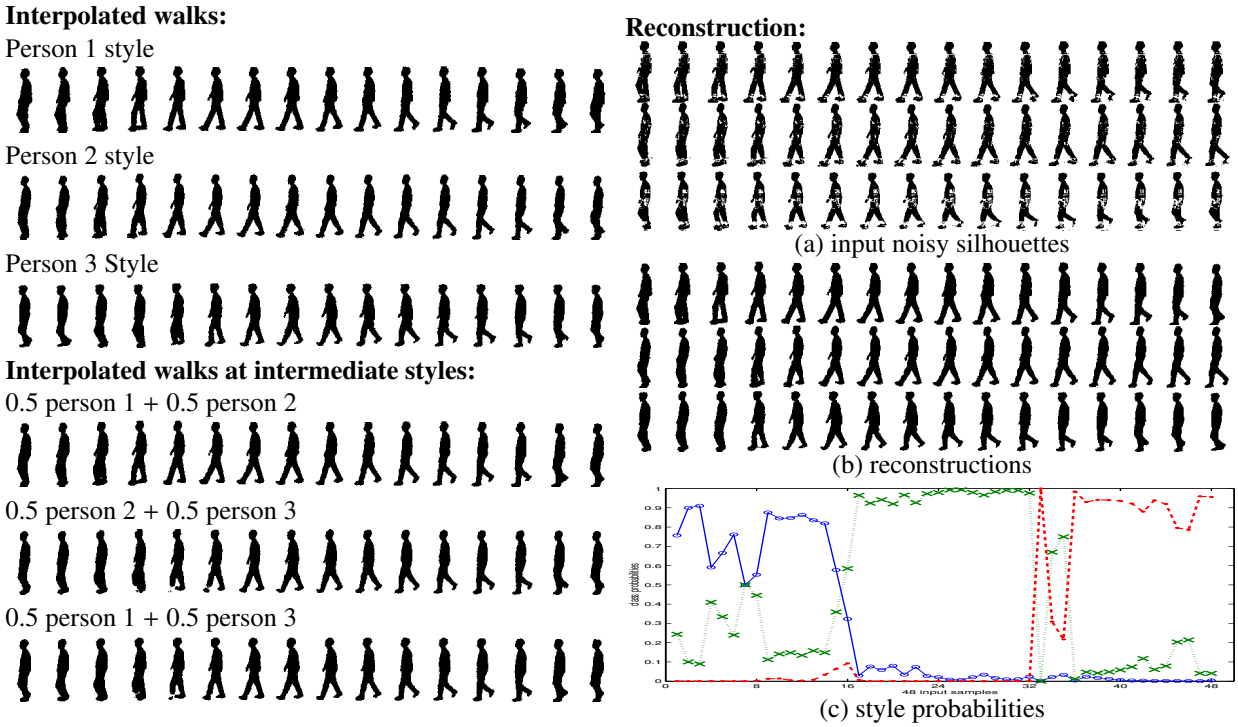


Figure 9: Left: Interpolated walks at different people shape styles. Right: Reconstruction example. a) input noisy silhouettes. b) pose-preserving style-preserving reconstruction. c) estimated style probabilities.

and the resulting reconstructions are shown in Fig. 9-b in the same order. Notice that the reconstruction preserves both the correct body pose as well as the correct person shape. Only two errors can be spotted, which are for inputs number 33, 34 (last row, columns 2,3) where the probability for person 2 was higher than person 3, and therefore the reconstruction preserved the second person's style. Fig. 10 shows another reconstruction example, where the learned model was used to reconstruct corrupted inputs for person 3. The reconstruction preserves the person style, as well as the body pose.

7.2.3. Experiment 3 Shape and Gait Interpolation

In this experiment we fit the single style factor model in Eq. 4 with a larger data set. We used five sequences for five different people, each containing about 300 frames, which are noisy (extracted using background subtraction). The learned manifolds are shown in Fig. 11-a, which shows a different manifold for each person. The learned unified manifold is also shown in Fig. 11-b. Fig. 11-c shows in-

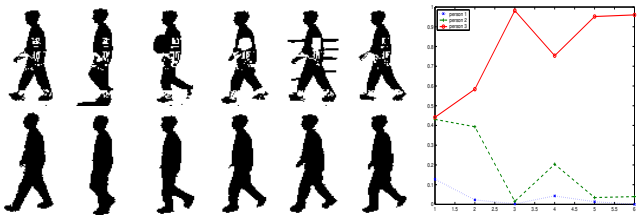


Figure 10: Pose and style preserving reconstruction. Right: style probabilities for each input

terpolated walking sequences for the five people generated by the learned model. The figure also shows the learned style vectors. We evaluated style classifications using 40 frames for each person and the results are shown in the figure with correct classification rate of 92%. We also used the learned model to interpolate walks in new styles. Fig. 12 shows interpolation between person 1 (big man) and person 4 (slim woman). We used linear interpolation in the style (shape) space and rendered the resulting silhouettes at different phases of the gait cycle. The interpolations successfully exhibit walking figures at intermediate shape styles between person 1 and 4.

7.3. A Multifactor Model for Gait

In this section we show experiments on fitting a multifactor model for gait, as described earlier in Section 3.2, and as illustrated in Fig. 3. The model decomposes the viewpoint and the shape as two style factors, while the gait motion is the content.

7.3.1. Evaluation on CMU MoboGait Dataset

We used five people, five cycles each, from four different views, from the CMU MoboGait dataset [54], i.e., the total number of cycles for training is 100 = 5 people $\times$ 5 cycles $\times$ 4 views. The number of frames in each cycle is different within the same person's cycles, as well as across different people. Fig. 3 shows examples of the sequences with different views (only half cycles are shown in the figure). The silhouette data used is noisy, as typically the result from background subtraction.

We learned the model in Eq. 5 using the collected 100 sequences. Images are normalized to 60×100 (width $\times$ height) i.e., $d = 6000$. In this experiment, a unit circle is used as a conceptual model for the gait manifold invariant to shape and view.

Each cycle is considered to be a style by itself, i.e., there are 25 styles and 4 views. 18 equidistant points on the unit circle were used to obtain the nonlinear

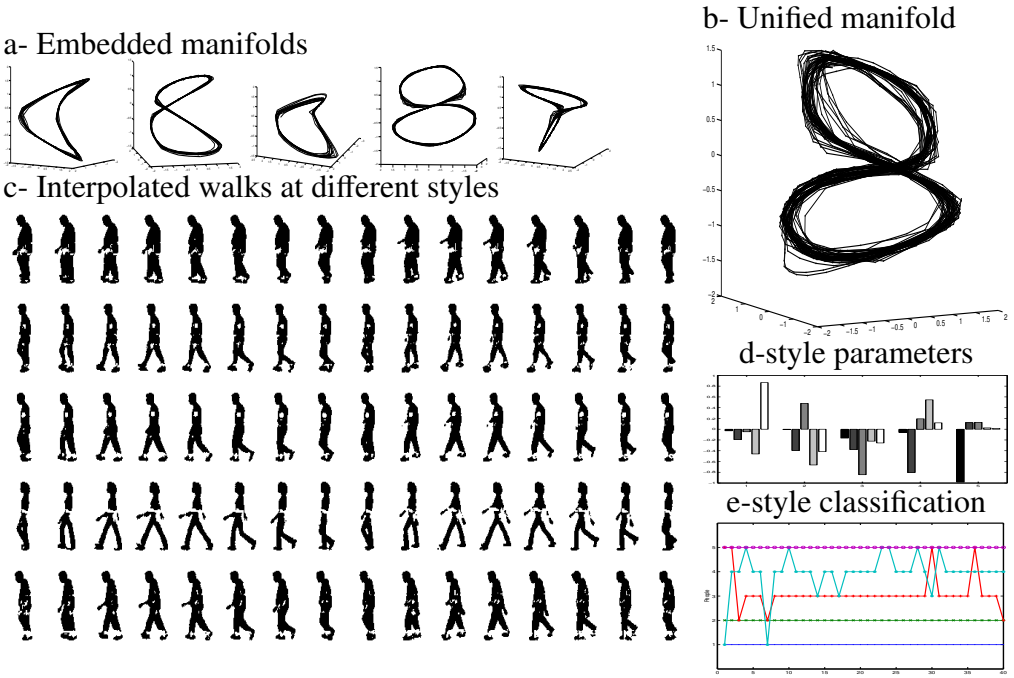


Figure 11: a- the learned manifolds for each of the five subjects. b- the learned unified content manifold. c- interpolated walks at each person style. d- Shape style vectors. e- shape style classification results - 40 frames used each row corresponds to one of the five subjects.

mapping defined in Eq. 9. After coefficient decomposition and dimensionality reduction, as in Eq. 12, the dimensions for core tensor $\mathcal{D}$ is $5 \times 4 \times 120$. The dimensions for the basis matrices are 25×5, 4×4, and $(18 \times 6000) \times 120$ for shape, view, and coefficient basis respectively. Fig. 13-b shows an example of a unit circle embedding of three cycles after alignment of cycles. Fig. 13-a shows the obtained style subspace, where each of the 25 points corresponding to one of the 25 cycles used. An important result to notice, is that the style vectors are clustered in the subspace such that each person's style vectors (corresponding to different cycles of the same person) are clustered together ?which indicates that the model preserves the similarity in the shape style between different cycles of the same person. Fig. 13-c shows the mean style vector for each of the five clusters. Fig. 13-d shows the four view vectors.

Gait Pose, Style, and View Estimation: In this experiment, we used the learned model to evaluate the recovery of body configuration, view, and person shape style. We used two new test cycles for each of the five people used in training from the four views, i.e., 40 cycles with a total of 1344 frames in all the test se-

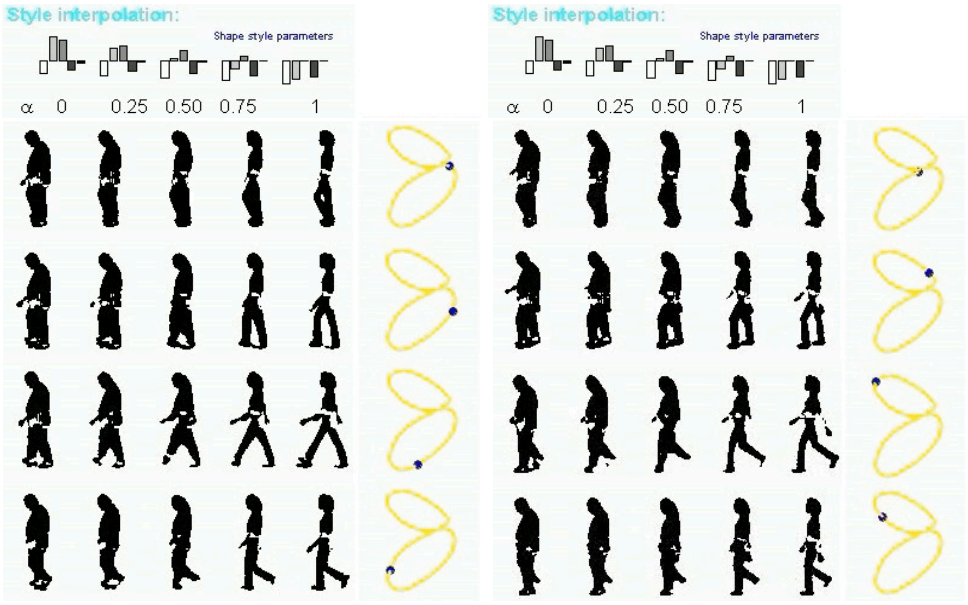


Figure 12: Linear interpolation in the shape space between two subjects and the resulting shapes at eight different points of the gait cycle

quences. If we use a whole cycle for recovery of view and person style parameter as described in 6.1, we obtain 100% correct view classification. For style classification, we get 36 out of 40 correct classification using nearest style mean and 40 out of 40 using nearest neighbor classifier. If we use individual frames for recovery, as described in Section 6.2, we get 7 frame errors amongst the 1344 test frames for body configuration and style estimation, i.e., 99.5% accuracy with 100% correct view estimation. In our experiment, a body configuration is considered an error if the angle between the correct and estimated embedding is more than $\pi/8$, which is about 2 to 4 frame difference in the original sequence.

Fig. 13-Right shows examples of using the model to recover the pose, viewpoint, and shape style. The figure shows samples of one full cycle and the recovered body configuration at each frame. Notice that despite the similarities between the first half and the second half of a cycle, the model exploits the subtle differences to recover the correct pose. The recovery of 3D joint angles is achieved by learning a mapping from the manifold embedding and 3D joint angle from motion captured data using GRBF in a way similar to Eq. 8. Fig. 14-a,b show the recovered style weights (class probabilities) and view weights respectively for each frame of the cycle which shows correct person and view classification. Fig. 14-c

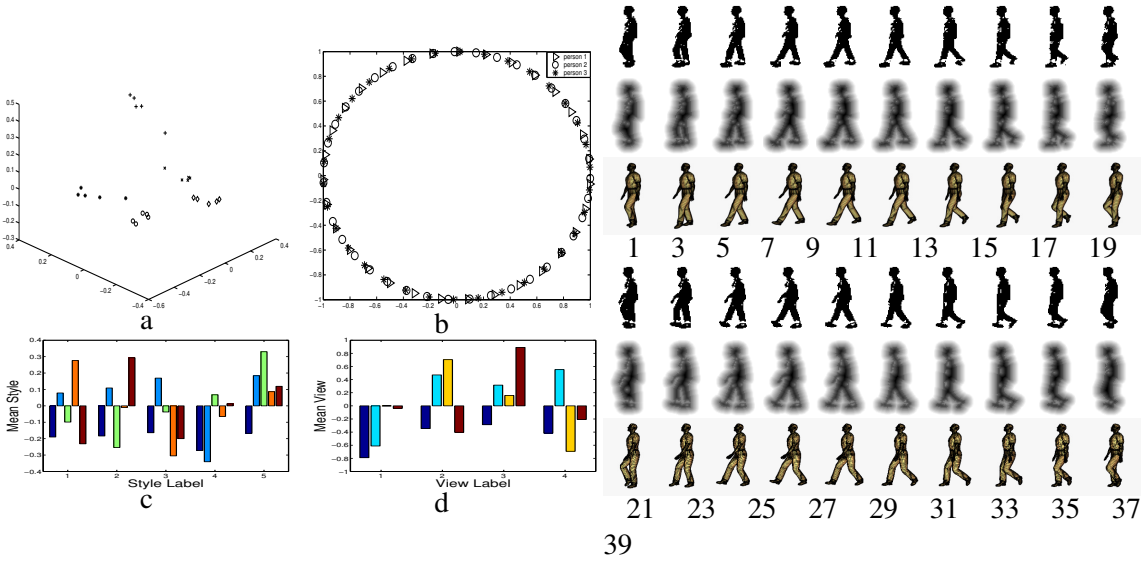


Figure 13: Left: Learned style and view vector. a) Style subspace: each person cycles have the same label. b) Unit circle embedding for three cycles. c) Mean style vectors for each person's cluster. d) Viewpoint vectors. Right. Example pose recovery. From top to bottom: input shapes, implicit function, and recovered 3D pose.

visualizes the progress of the error, style weights, and view weights, through the iterations used to obtain the results for frame 5. As can be noticed, the weights start uniformly and then smoothly converge into the correct style and view as the error is reduced and the correct body configuration is recovered.

Generalization to New Subjects: In this experiment we used the learned model to evaluate the recovery of body configuration and view, using test data for new subjects. We used 8 people sequences, 2 cycles each, from 4 views, where none of them were used in the training. Overall, there are 2476 frames in the test sequences. The recovery of the parameters was done on a single frame basis, as described in section 6.2. We obtained 111 errors in the recovery of the body configuration, i.e., body configuration accuracy is 95.52%. Error in body configuration is measured in the embedding space using the same way described earlier. For view estimation we get 7 frame errors, i.e., view estimation accuracy 99.72%. Fig. 15 shows examples of recovery of the 3D pose and viewpoint.

7.3.2. Evaluation on HumanEva Dataset

We evaluated the multifactor gait model on the HumanEva dataset [55], which is a commonly used benchmark for evaluating pose estimation algorithms. The

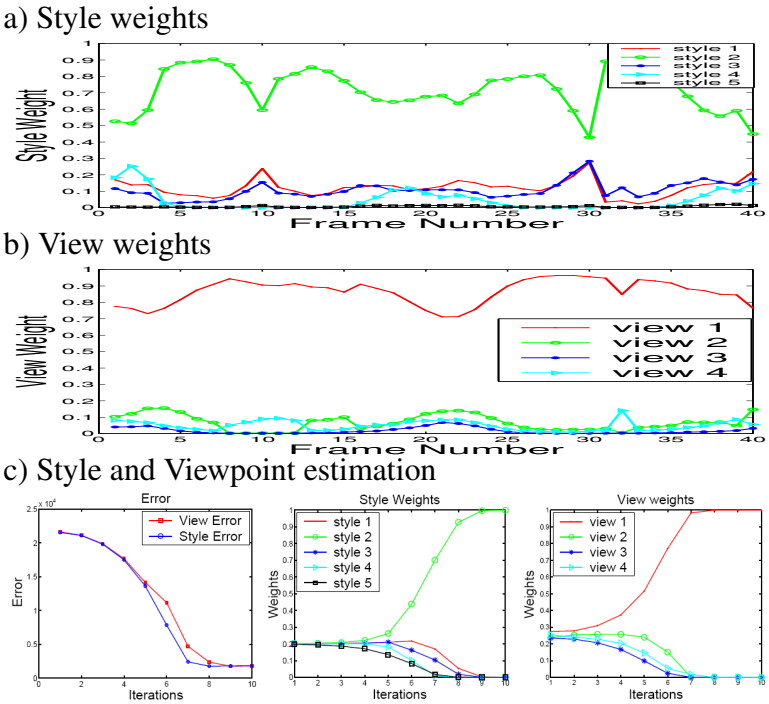


Figure 14: Estimated weights during a cycle a) Style weights. b) View weights. c) Iterative style and view estimations for each frame. Left: Error. Center: style weights. Right: View weights

dataset consists of two subsets: HumanEva-I, which contains sequences of four subjects performing six actions; and HumanEva-II, which contains sequences of two subjects performing combination of actions. The dataset contains videos from four to seven cameras, as well as motion-captured ground truth data. HumanEva-I contains data for training, validation, and testing; while Human-Eva-II contains two sequences for testing only. HumanEva-II video data is hardware synchronized, therefore approaches that use multiple cameras typically prefer to test on HumanEva-II. For details we refer the reader to [55].

As mentioned in Section 4, there are different ways to achieve the content embedding. We evaluated two variants of the model with two different content embedding: 1) using supervised conceptual embedding on a unit circle, 2) using embedding from kinematic data (auxiliary data). We used synthetic walking sequence captured from 12 viewpoints to train the models. The factorized viewpoint variable, in Eq 5, is used fit a one-dimensional spline curve representing the viewpoint manifold in the factorized view subspace. For evaluation of the 3D reconstruction accuracy, we learn a nonlinear mapping from the content embedding to

Figure 15: Examples of pose recovery and view classification for three people.

the 3D motion-captured joint locations. One cycle of the training motion-captured data in HumanEva-I dataset is used to learn that mapping for each subject.

For testing, we used background subtraction to segment the subject's silhouettes. For continuous tracking, a particle filter was used to solve the inference problem. Once the configuration variable x_t is recovered, the mapping from the content embedding to the 3D joint location space is used to recover the 3D pose. The mean-squared error between the estimated joint locations (relative to a body centric coordinate system) and the ground truth is used to evaluate the performance.

We used subjects S1 and S2 from HumanEva-I subset for the evaluation. We used the same number of particles, same initialization, and the same dynamic model on the content manifold for all the compared realizations of the model. We tested using sequences from a single camera (camera 2). Table 1 illustrates the results for the two realizations of the model. The results of the two realizations are comparable, with the unit circle embedding giving slightly better estimation. This is expected since the unit-circle embedding provides a more stronger prior than the kinematic embedding for the case of walking motion. Figure 16 shows sample of the input silhouettes, the reconstructed shapes (after estimating the pose and viewpoint), and the estimated 3D pose; using the unit-circle embedding. Figure 17-a shows the errors in estimation per-frame for subject S1 for each realization. Figure 17-b shows an example of the estimated joint angle location (X-axis of the lower left leg distal) compared to the ground truth.

Table 2 shows the results of several pose estimation approaches on HumanEva dataset, in comparison to the proposed framework. We mainly selected the approaches that reported results on walking sequences of the HumanEva datasets. It is very hard to compare reported results in HumanEva dataset since different authors use different subsets, different sequences, different parts of the sequences, different number of cameras, and even different error metrics in their evaluation. Since the metric used is the average error per frame, we report the average error for each approach over the different used subsets, and we also report the subsets

(a) Input

(b) Reconstruction

(c) Estimated 3D Posture

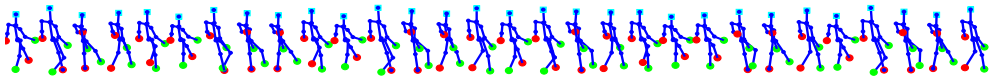


Figure 16: Sample from walking sequence S1 HUMANEVA-I: (a) Input silhouettes. (b) Synthesized silhouettes after viewpoint and body configuration estimation. (c) Reconstructed 3D postures.

used. As can be seen from the table, the proposed approach gives the best pose estimation results from a single camera. The error in pose estimation is even better than most approaches that use multiple cameras.

7.4. A Multifactor model for Facial Expression Analysis

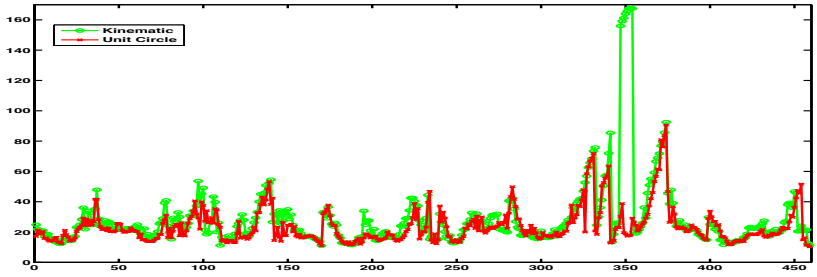
7.4.1. Facial Expression Synthesis and Recognition

We learned and evaluated the model in Eq 6 using different datasets. Here we report results using the Cohn-Kanade AU coded facial expression database [56]. We used eight subjects with all six basic emotions (anger, disgust, fear, joy, sadness, surprise), that is 48 expression sequences with varying number of frames per sequence (between 11 and 33) . Each sequence represents an expression starting from neutral to expression peak. We used a unit circle as the motion manifold

Table 1: Average mean squared errors the 3D body joint position estimation (relative to a body-centered coordinate system) for HumanEva I dataset from a single camera

Subject	**Start**	**End**	**Duration**	**Cycles**	**Kinematic Embedding**	**Conceptual Unit Circle Embedding**
S1	76	534	459	6	28.37 *mm*	23.84 *mm*
S2	21	436	416	5	26.68 *mm*	26.81 *mm*
Average			407.6	5.5	27.53 *mm*	25.33 *mm*

(a)



(b)

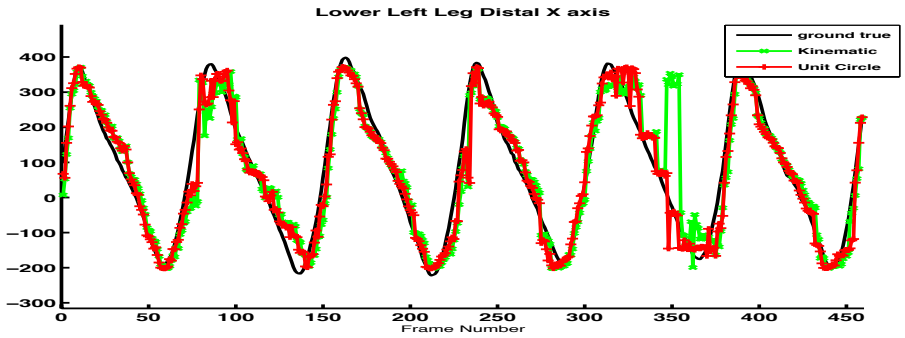


Figure 17: Results from HumanEva I-S1 (a) Joint location error in each frame in *mm*. (b) Details of lower left leg distal reconstruction(X axis).

representation[6]. Sequences were spatially aligned based on eye and nose template alignment. Eight equidistant centers were used in learning GRBF with thin-plate spline basis. We used the full dimensions to represent each style (8) and expression (6). Fig. 18 shows representation of expression vectors and personal style vectors (facial appearance) after learning the model. It is interesting to notice that the anger, fear, disgust, and sadness expression vectors are close to each other in the expression space, while the surprise and smile are further away. Fig 20 shows the use of the learned model to generate faces of different people (along the rows) with different expressions (along the columns).

Sequence-based expression recognition: The performance of person independent facial expression recognition is tested by leave-one-out cross-validation. For

[6]The sequences were embedded to half a unit circle and the reverse of the sequences were embedded to the other half of the circle.

Table 2: Comparison to stat-of-the-art approaches using HumanEva dataset

Approach	Number of Cameras	Average Error (mm)	Sequences used
baseline [55]	1	520.5 [a]	II-S2, II-S4
Xu et al. [57]	1	148.67 [r]	I-S1, I-S2, I-S3
Andriluka et al [58]	1	104 [r]‡	II-S2
Brubaker et al [59]	1	70.75 [r]‡	II-S2, II-S4
Elgammal et al. [50]	1	31.36 [r]	I-S1, I-S2 I-S3
HMA- Kinematic	1	**27.53** [r]	I-S1, I-S2
HMA- Unit Circle	1	**25.33** [r]	I-S1, I-S2
baseline [55]	2	135.5 [a]	II-S2, II-S4
baseline [55]	3	69.5 [a]	II-S2, II-S4
baseline [55]	4	68.0 [a]	II-S2, II-S4
Gall et al. [60]	4	34.75 [r]	II-S2, II-S4 *
Raskin et al. [61]	4	75.4 [a]	I-S1
Corazza et al. [62]	4	75.5 [am]	II-S2, II-S4
Cheng et al [63]	4	15.5 [r]	II-S2, II-S4 *

[a] Absolute joint locations are used in measuring the error. [r] Relative joint locations, w.r.t. to a body centered coordinate system, are used. ‡ Average of two results from two different cameras. [m] First 150 frames were used. * Results for walking only are considered.

this purpose we learned the model using 42 sequences of seven subjects, and tested on the six sequences of the eighth subject. After solving for the expression and facial appearance factors using the procedure described in Sec 6.1, the estimated expression vector is used for classification using a nearest neighbor classifier. Table 3 shows the confusion matrix for the 48 test sequences.

Frame-based expression recognition: Using the learned generative model as described above, we can estimate person face and expression parameters from each single frame using the deterministic annealing procedure described in Sec. 6.2. We used 16 additional subjects (not used for training) from the same dataset with five expressions each. The expressions for each subject varies. Fig. 19 shows the expression weight values α's of every frame in two different sequences. The weights become more discriminative as the frames get closer to the peak of the expressions. The expression with the maximum weight is used as the classification result. Table 4 shows recognition results using the estimated weights at the last frame of each sequence.

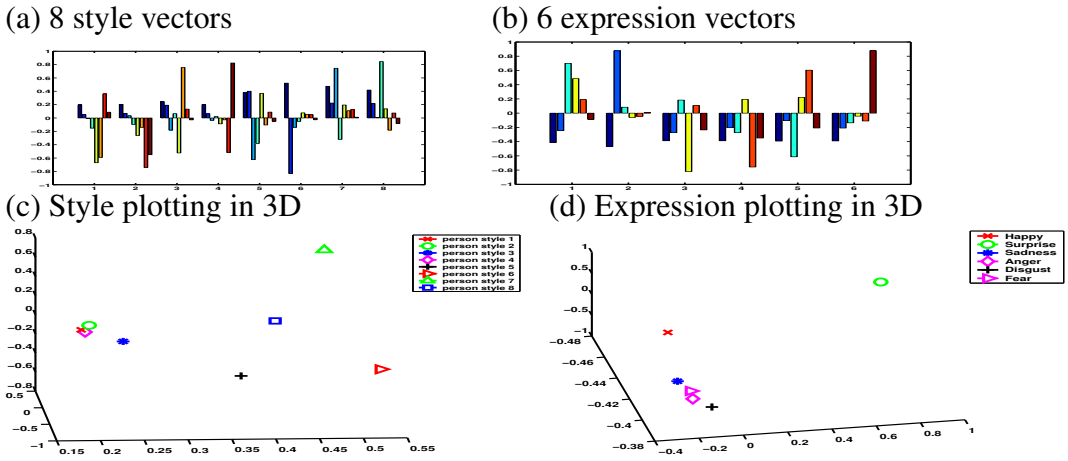


Figure 18: Facial expression analysis for Cohn-Kanade Dataset for 8 subjects with 6 expressions and their 3D space plotting

Table 3: Confusion matrix for Sequence-based facial expression recognition.

Emotion	Joy	Surprise	Sadness	Anger	Disgust	Fear
Joy	25%(2)	0	0	25%(2)	25%(2)	25%(2)
Surprise	12.5%(1)	62.5%(5)	12.5%(1)	0	0	12.5%(1)
Sadness	0	0	37.5%(3)	25%(2)	12.5%(1)	25%(2)
Anger	12.5%(1)	0	37.5%(3)	50%(4)	0	0
Disgust	12.5%(1)	12.5%(1)	12.5%(1)	25%(2)	12.5%(1)	25%(2)
Fear	0	0	0	50%(4)	0	50%(4)

7.5. Facial Expression Synthesis

The generative model can be used to render facial animations by controlling the different variables (motion phase, expression variable, face appearance variable). Convex linear combination between learned facial appearance styles and different facial expressions can be used to interpolate between the faces and expressions. Although linear interpolation is used in the space of each variable, the resulting animation will show nonlinear facial deformations because of the nonlinear manifold mapping. Fig. 21 shows different examples of facial synthesis, including transition between expressions during a motion cycle, transition between faces during an expression, and transition between faces and expressions simultaneously (see caption for details).

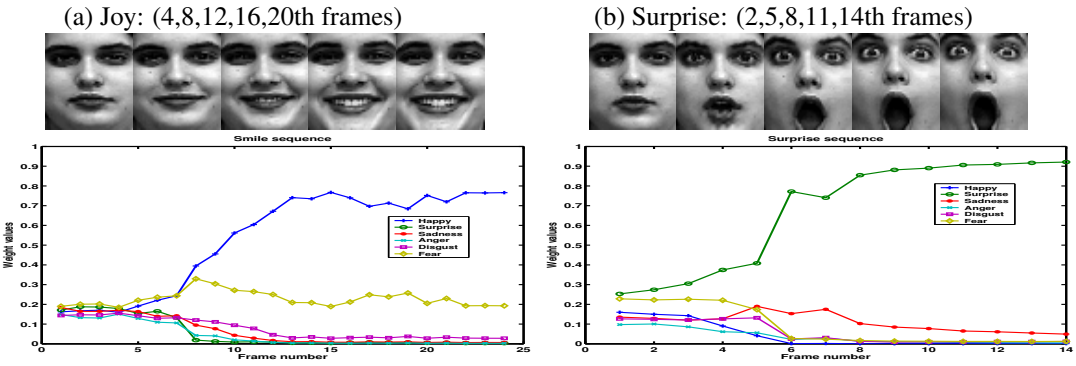


Figure 19: Estimated expression weights using frame-based estimation.

Table 4: Confusion matrix for frame-based recognition: classification only at last frame (peak)

Emotion	Joy	Surprise	Sadness	Anger	Disgust	Fear
Joy	93.3%(14)	0	0	0	0	6.7%(1)
Surprise	0	100%(16)	0	0	0	0
Sadness	0	7.1%(1)	28.6%(4)	7.1%(1)	35.7%(5)	21.4%(3)
Anger	9.1%(1)	0	18.2%(2)	27.3%(3)	45.4% (5)	0
Disgust	9.1%(1)	0	9.1%(1)	18.2%(2)	63.6%(7)	0
Fear	24.9%(3)	0	8.3%(1)	0	8.3%(1)	58.3%(7)

8. Conclusions and Discussion

We introduced a framework for separating style and content on manifolds representing dynamic objects. The framework is based on factorizing style variables in the space of nonlinear functions that maps between a unified nonlinear embedding of the content manifold and style-dependent observations in the visual input space. We introduced three different methodologies to obtain a unified content manifold embedding: 1) through unsupervised nonlinear dimensionality of visual data and manifold warping, 2) through supervised conceptual embedded representation of the manifold; 3) through nonlinear dimensionality reduction of auxiliary data (e.g. motion-captured data) that is invariant to the visual variability. As mention in [21], an interesting and important question is how to learn a parametric mapping between the observation and nonlinear embedding spaces. We addressed this question in this paper. The proposed framework is not tied to a specific embedding approach. Any nonlinear dimensionality reduction approach can be used to obtain manifold embeddings, which can be then used to learn the model.

The proposed framework was shown to be able to separate content and mul-

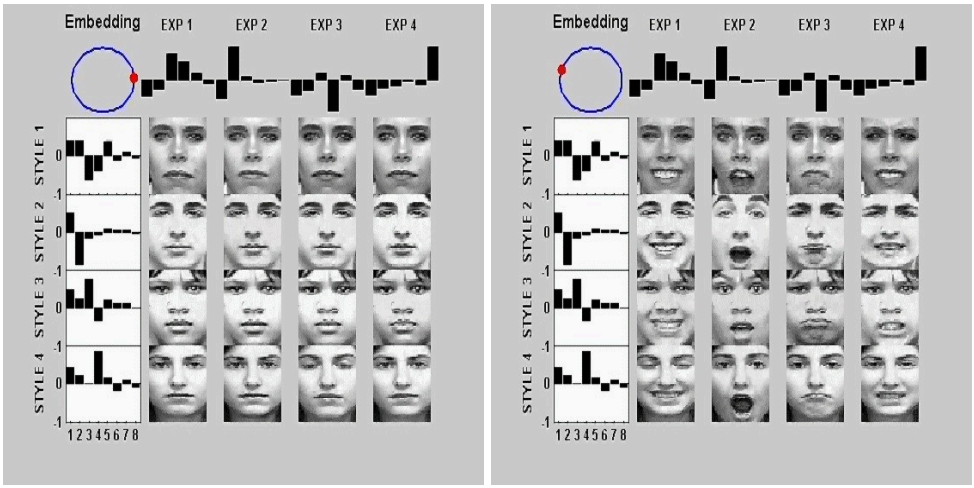


Figure 20: Factorized Facial Expression Model: using the learned model to generate faces of different people (along the rows) with different expressions (along the columns). Only four expressions are shown (Happy, Surprise, Sadness, Disgust). Moving along the unit-circle manifold represent the motion phase of all expressions, from neutral to peak to neutral. Two phases are shown, neutral and peak

tiple style factors for the gait and facial expression manifolds. We showed three applications of the framework: 1) a single factor model for gait or individual facial expressions, 2) a multifactor model for gait, 3) a multifactor model for facial expressions. For all the cases, experiments showed the applicability of the model, and very good results for synthesis and parameter recovery. We achieved the state-of-the-art results for posture estimation for walking motion, evaluated on the HumanEva [55] benchmark dataset from a single camera.

One of the features of the proposed framework is that the separation of style is within a generative model. The use of a generative model is tied to the use of a manifold embedding, since the mapping from the manifold representation to the input space will be well-defined. This is in contrast to a discriminative mapping from the visual input to an embedded manifold representation, which is not necessarily a function. Since the framework is generative, it is suitable for the Bayesian tracking framework and it provides separate low-dimensional representations for each of the modeled factors. Moreover, a dynamic model for body configuration can be defined on the manifold representation. We investigated the use of the model with Bayesian tracking in [43, 45, 50].

We introduced an optimization framework to solve for the various factors, such as body configuration, view, and shape style. We showed that we can solve for

(a) Neutral → smile → surprise (b) Surprise → angry → neutral

(c) Subject A face → subject B face (d) Subject B face → subject C face

(e) Simultaneous transfer of face and expression (neutral → smile → surprise → fear → neutral while warping between Subject A to B to C)

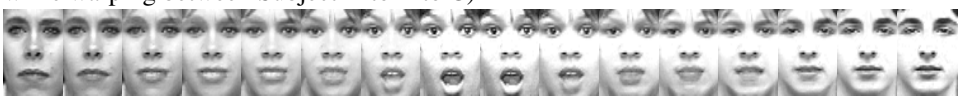


Figure 21: Facial expression synthesis: First row: Transition between expressions: a) from neutral to 50-50% smile-surprise to 100% surprise (at peak). b) From peak with100% surprise to 50-50 surprise-anger to neutral. Second row (c,d) : Transition between faces during smile expressions. Third row: simultaneous transition of facial expression and person faces.

all these factors given a single input image or a whole motion cycle. Therefore, the framework provides a way to initialize a tracker by inferring the body configuration, viewpoint, and body shape style from a single or a sequence of images. Since the model is generative, various sampling techniques, such as MCMC and Particle Filters, can be also used for inferring the different variables. We showed results using different inference approaches.

The framework presented in this paper was applied to one-dimensional motion manifolds such as gait and facial expressions. One-dimensional manifolds can be explicitly modeled in a straightforward way. We envision that a complex motion can always be segmented into short motions that are restricted to one-dimensional manifolds. However, there is no theoretical restriction that prevents the framework from dealing with more complicated manifolds. In [43] we used the framework to model more complex actions, such as ballet motions, aerobic dance, etc.

In this paper, we primarily modeled the motion explicitly as a manifold, while all appearance variability are modeled using multilinear subspace analysis. A style factor can also be modeled as a manifold in its factorized subspace, if dense samples are available. In the experiment with the HumanEva dataset, the viewpoint manifold was parameterized as a configuration-invariant manifold in the subspace resulting from the factorization. This way we model both the configuration and viewpoint manifolds explicitly.

Modeling data lying on a combination of manifolds can also be achieved using the proposed framework if the combined manifold topology is known. The

idea of using a supervised conceptual model for complex manifolds was further developed in [50], to model both the body configuration and viewpoint manifolds simultaneously for gait motion using a torus manifold.

The framework presented in this paper assumes that the inputs are represented in a Euclidean space. We mainly used shapes and appearances represented as vectors. It is very interesting and challenging to carry this framework to other input representations such as sparse features, e.g., SIFT [64] features. The problem in this case is how to represent such sparse features in a Euclidian space. The framework presented here is orthogonal to such a representation issue. In [65] we introduced an approach for learning image manifolds from sparse local features that takes into consideration both the feature appearance and their spatial arrangement. Such an approach can be used in conjunction with the proposed framework to learn a generative model for sparse features.

References

[1] H Sebastian Seung and Daniel D. Lee, "The manifold ways of perception," *Science*, vol. 290, no. 5500, pp. 2268–2269, 2000.

[2] J. Tenenbaum, "Mapping a manifold of perceptual observations," in *Proc. of NIPS*, 1998, vol. 10, pp. 682–688.

[3] Richard Bowden, "Learning statistical models of human motion," in *Proc. IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000, pp. 10–17.

[4] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *Proc. of CVPR*, 2004, vol. 2, pp. 681–688.

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

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[7] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," in *Proc. of ECCV*, 1996, pp. 45–58.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models: Their training and applications," *CVIU*, vol. 61, no. 1, pp. 38–59, 1995.

[9] A. Levin and A. Shashua, "Principal component analysis over continuous subspaces and intersection of half-spaces," in *Proc. of ECCV*, 2002, pp. 635–650.

[10] H. Murase and S. Nayar, "Visual learning and recognition of 3d objects from appearance," *IJCV*, vol. 14, no. 1, pp. 5–24, 1995.

[11] Joshua B. Tenenbaum and William T. Freeman, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, pp. 1247–1283, 2000.

[12] M. Alex O. Vasilescu and Demetri Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. of ECCV*, 2002, pp. 447–460.

[13] Jan R. Magnus and Heinz Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons, 1988.

[14] David H. Marimont and Brian A. Wandell, "Linear models of surface and illuminant spectra," *Journal of Optical Society of America*, vol. 9, no. 11, pp. 1905–1913, 1992.

[15] Lieven De Lathauwer, Bart de Moor, and Joos Vandewalle, "A multilinear singular value decomposiiton," *SIAM Journal On Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[16] A. Shashua and A. Levin, "Linear image coding of regression and classification using the tensor rank principle," in *Proc. of CVPR*, 2001.

[17] M. Alex O. Vasilescu, "Human motion signatures: Analysis, synthesis, recogntion," in *Proc. of ICPR*, 2002, vol. 3, pp. 456–460.

[18] Ledyard R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[19] A. Kapteyn, H. Neudecker, and T. Wansbeek, "An approach to n-model component analysis," *Psychometrika*, vol. 51, no. 2, pp. 269–275, 1986.

[20] R. Fablet and M. J. Black, "Automatic detection and tracking of human motion with a view-based representation," in *Proc. of ECCV*, 2002, pp. 476–491.

[21] Sam Roweis and Lawrence Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[22] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[23] M. Brand and K. Huang, "A unifying theorem for spectral embedding and clustering," in *In Proc. of the Ninth International Workshop on AI and Statistics*, 2003.

[24] N. Lawrence, "Gaussian process latent variable models for visualization of high dimensional data," in *Proc. of NIPS*, 2003.

[25] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-Francois Paiement, Pascal Vincent, and Marie Ouimet, "Learning eigenfunctions links spectral embedding and kernel pca," *Neural Comp.*, vol. 16, no. 10, pp. 2197–2219, 2004.

[26] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of ICML*, 2004, p. 47.

[27] Bernhard Schlkopf and Alex Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2002.

[28] 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," in *Proc. of NIPS*, 2004.

[29] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proc. of ICCV*, 2001, pp. 50–59.

[30] C. Bregler and S. M. Omohundro, "Nonlinear manifold learning for visual speech recognition," in *Proc. of ICCV*, 1995, pp. 494–499.

[31] M. Brand, "Shadow puppetry," in *Proc. of ICCV*, 1999, vol. 2, pp. 1237–1244.

[32] D. Ormoneit, H. Sidenbladh, M. J. Black, T. Hastie, and D. J. Fleet, "Learning and tracking human motion using functional analysis," in *Proc. IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000, pp. 2–9.

[33] Cristian Sminchisescu and Allan Jepson, "Generative modeling for continuous nonlinearly embedded visual inference," in *Proceedings of ICML*. 2004, pp. 96–103, ACM Press.

[34] Ali Rahimi, Ben Recht, and Trevor Darrell, "Learning appearance manifolds from video," in *Proc. of CVPR*, 2005, vol. 1, pp. 868–875.

[35] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua, "Priors for people tracking from small training sets," in *Proc. of ICCV*, 2005, pp. 403–410.

[36] Vlad I. Morariu and Octavia I. Camps, "Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics," in *Proc. of CVPR*, 2006, pp. 545–552.

[37] Kooksang Moon and Vladimir Pavlovic, "Impact of dynamics on subspace embedding and tracking of sequences," in *Proc. of CVPR*, 2006, vol. 1, pp. 198–205.

[38] Raquel Urtasun, David J. Fleet, and Pascal Fua, "3d people tracking with gaussian process dynamical models," in *Proc. of CVPR*, 2006, pp. 238–245.

[39] Jack Wang, David J. Fleet, and Aaron Hertzmann, "Gaussian process dynamical models.," in *Proc. of NIPS*, 2005.

[40] S. Roweis and Z. Ghahramani, "An EM algorithm for identification of nonlinear dynamical systems," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed.

[41] Tai-Peng Tian, Rui Li, and Stan Sclaroff, "Articulated pose estimation in a learned smooth space of feasible solutions," in *Proc. of CVPR*, 2005, p. 50.

[42] Chris Mario Christoudias and Trevor Darrell, "On modelling nonlinear shape-and-texture appearance manifolds.," in *Proc. of CVPR*, 2005, pp. 1067–1074.

[43] Chan-Su Lee and Ahmed Elgammal, "Modeling view and posture manifolds for tracking," in *Proc. of ICCV*, 2007.

[44] G. S. Kimeldorf and G. Wahba, "A correspondence between bayesian estimation on stochastic processes and smoothing by splines.," *The Annals of Mathematical Statistics*, vol. 41, pp. 495–502, 1970.

[45] Chan-Su Lee and Ahmed Elgammal, "Style adaptive bayesian tracking using explicit manifold learning," in *Proc. of British Machine Vision Conference*, 2005, pp. 739–748.

[46] Ahmed Elgammal and Chan-Su Lee, "Nonlinear manifold learning for dynamic shape and dynamic appearance," *CVIU*, vol. 106, no. 1, pp. 31–46, 2007.

[47] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Proc. of CVPR*, 2000, pp. 44–51.

[48] M. Torki, A. Elgammal, and C-S. Lee, "Learning a joint manifold representation from multiple data sets," in *Proc. of ICPR*, 2010.

[49] A. Elgammal, "Learning to track: Conceptual manifold map for closed-form tracking," in *Proc. of CVPR*, June 2005.

[50] Ahmed Elgammal and Chan-Su Lee, "Tracking people on a torus," *IEEE Trans. PAMI*, pp. 520–531, march 2009.

[51] Tomaso Poggio and Fredrico Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[52] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, , no. 33, pp. 8295, 1971.

[53] Lieven De Lathauwer, Bart de Moor, and Joos Vandewalle, "On the best rank-1 and rank-(r1, r2, ..., rn) approximation of higher-order tensors," *SIAM Journal On Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[54] R. Gross and J. Shi, "The cmu motion of body (mobo) database," Tech. Rep. TR-01-18, Carnegie Mellon University, 2001.

[55] Leonid Sigal and Michael J. Black, "Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion," Tech. Rep. CS-06-08, Brown University, 2006.

[56] Takeo Kanade, Yingli Tian, and Jeffrey F. Cohn, "Comprehensive database for facial expression analysis," in *Proc. of FGR*, 2000, pp. 46–53.

[57] Xinyu Xu and Baoxin Li, "Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter," in *Proc. of ICCV*, 2007, pp. 1–8.

[58] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *Proc. of CVPR*, june 2010, pp. 623 –630.

[59] Marcus A. Brubaker, David J. Fleet, and Aaron Hertzmann, "Physics-based person tracking using the anthropomorphic walker," *IJCV*, vol. 87, no. 1-2, pp. 140–155, Mar. 2010.

[60] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel, "Optimization and filtering for human motion capture," *IJCV*, vol. 87, no. 1-2, pp. 7592, 2010.

[61] Leonid M. Raskin, Michael Rudzsky, and Ehud Rivlin, "3d human body-part tracking and action classification using a hierarchical body model," in *British Machine Vision Conference*, 2009.

[62] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *IJCV*, vol. 87, no. 1-2, pp. 156–169, 2010.

[63] Shinko Y. Cheng and Mohan M. Trivedi, "Articulated human body pose inference from voxel data using a kinematically constrained gaussian mixture model," .

[64] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[65] M. Torki and A. Elgammal, "Putting local features on a manifold," in *Proc. of CVPR*, 2010.