MODELING HUMAN MOTION USING MANIFOLD LEARNING AND FACTORIZED GENERATIVE MODELS

BY CHAN-SU LEE

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

written under the direction of

Ahmed Elgammal

and approved by

New Brunswick, New Jersey

May, 2007

ABSTRACT OF THE DISSERTATION

Modeling Human Motion Using Manifold Learning and Factorized Generative Models

by Chan-Su Lee

Dissertation Director: Ahmed Elgammal

Modeling the dynamic shape and appearance of articulated moving objects is essential for human motion analysis, tracking, synthesis, and other computer vision problems. Modeling the shape and appearance of human motion is challenging due to the high dimensionality of the articulated human motion, variations of shape and appearance from different views and in different people, and the nonlinearity in shape and appearance deformations in the observed sequences. Recent interest in modeling human motion is originated from the various potential real-world applications such as visual surveillance, human-computer interaction, video analysis, computer animation, etc.

We present a novel framework to model dynamic shape and appearance using nonlinear manifold embedding and factorization. We investigate different representations to embed high-dimensional human motion sequences in low dimensional spaces by supervised and unsupervised manifold learning techniques to achieve representations that capture the intrinsic structure of the motion. Nonlinear dimensionality reduction techniques based on visual data and kinematic data are applied to discover low dimensional intrinsic manifold representation for body configuration. Also, we investigate the use of supervised manifold learning from a known manifold topology to model deformation of manifolds from an ideal case. By learning nonlinear mapping from the embedding space to the input shape or appearance, we can generate shape and appearance sequences according to the motion state on the embedded manifold.

We present a decomposable generative model to analyze shape and appearance variations by different factors such as person's style, motion type, and view point. We use multilinear analysis in the nonlinear mapping coefficient space to factorize shape and appearance variations. Also, we investigate learning generative models to represent continuous body configuration and continuous view manifolds in a product space (i.e. *body configuration manifold* $\times$ *view manifold*). The proposed factorized generative models provide rich models for the analysis of dynamic shape and appearance of human motion. We applied the model in computer vision problems such as inferring 3D body pose from 2D images, tracking human motion with continuous view variations within the Bayesian framework, and gait recognition. We also applied our model for facial expression analysis, tracking, recognition and synthesis.

Acknowledgements

I would like to sincerely thank Prof. Ahmed Elgammal, the chair of this thesis. It was such a privilege to have him as my advisor. Throughout my doctoral work, he encouraged me to develop research skills. He continuously stimulated my analytical thinking and greatly assisted me with scientific writing. I would also like to thank the other committee members of my dissertation, Prof. Dimitris Metaxas, Prof. Vladimir Pavlovic, and Dr. Jan Neumann for their advices and comments.

I would like thank Prof. Dimitris Samaras and Yang Wang at Stony Brook University for sharing high resolution facial expression data.

I'd like to thank my family. I'm grateful to my mother for her encouragement. I'm especially grateful to my wife. She provided comfort with love and allowed me to focus on my research. My son, Seungchan and my daugher, Hayoung make me happy along the way.

Dedication

To my family

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAM	is for active appearance model
ASM	is for active shape model
GRBF	is for generalized radial basis function
HMM	is for hidden Markov model
HOSVD	is for higher order singular value decomposition
Isomap	is for isometric feature map
LLE	is for locally linear embedding
PCA	is for principal components analysis
SVD	is for singular value decomposition
SVM	is for support vector machine
RBF	is for radial basis function
TPS	is for thin-plate spline

Chapter 1

Introduction

1.1 Motivation

In the last decade, extensive research has been performed to analyze, understand and recognize human motion from image sequences and from motion captured data. This wide interest originated from various potential real-world applications such as visual surveillance, human-machine interface, video archival and retrieval, computer graphics animation, autonomous driving, virtual reality, etc. The focus of the research covered a wide range of problems related to the analysis and synthesis of human motion including detection and tracking humans and their body parts, recovering body pose, understanding human motion and activities, recognizing gestures, and recognizing and synthesizing facial expressions. The analysis also addressed human identification based on face recognition, gait analysis, and other biometrics. Researchers have actively explored many computational models and machine-learning techniques for better understanding and modeling of human motions. However, state of the art computer vision systems can capture, analyze, and understand very limited motions in constrained environments. Still, variations of style in different people, view changes, and different motion complexity and dynamics cause a significant performance degradation in real-world applications.

1.1.1 Dimensionality of Articulated Human Motion

The human body is an articulated object with high degrees of freedom. The human body moves through the three-dimensional world and such motion is constrained by body dynamics and projected by lenses to form the visual input we capture through our cameras. Therefore, the changes (deformation) in appearance (texture, contours, edges, etc.) in the visual input (image sequences) corresponding to performing certain actions, such as facial expression or gesturing, are well constrained by the 3D body structure and the dynamics of the action being performed.

Despite the high dimensionality of the configuration space, many human motion activities lie intrinsically on low dimensional manifolds. This is true if we consider the body kinematics, as well as if we consider the observed motion through image sequences. Let us consider the observed motion. The shape of the human silhouette walking (e.g., Fig. 1.1) or performing a gesture is an example of a dynamic shape where the shape deforms over time based on the action performed. These deformations are constrained by the physical body constraints and the temporal constraints posed by the action being performed. If we consider these silhouettes through the walking cycle as points in a high dimensional visual input space, then, given the spatial and the temporal constraints, it is expected that these points will lay on a low dimensional manifold.

Similarly, the appearance of a face performing facial expressions is an example of a dynamic appearance that lies on a low dimensional manifold in the visual input space. In fact, if we consider certain classes of motion such as gait, a single gesture, or a single facial expressions, and if we factor out all other sources of variability, each of these motions lies on a one-dimensional manifold, i.e., a trajectory in the visual input space. Such manifolds are nonlinear and non-Euclidean.

1.1.2 Biological Motivation

Researchers have tried to exploit the manifold structure as a constraint in tasks, such as tracking and activity recognition, in an implicit way. While the role of manifold representations is still unclear in perceptions, it is clear that images of the same objects 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 activity firing is typically a function of a small number of variables, which implies that population activity also lie on low dimensional manifolds [114]. On the other hand, human visual perception shares representation with a motor control signal like mirror neurons [15]. Researchers also found that complicated motions can be described based on basic motion primitives [59]. Therefore, a manifold-based representation that connects kinematics with the visual input through learning some activity primitives is a biologically justified approach.

1.2 Our Approach: Nonlinear Factorized Generative Models

Considering an example of dynamic shape of human motion, we examine characteristics of dynamic shape and appearance of human motion. Then, we propose *nonlinear factorized generative models* using manifold embedding and factorization to achieve robust modeling of dynamic shape and appearance.

1.2.1 Characteristics of Dynamic Shape in Walking Sequence

The shape of the human silhouette through a walking cycle is an example of a dynamic shape where the shape deforms over time based on the action performed. These deformations are constrained by the physical body constraints and the temporal constraints posed by the action being performed. Dynamic shape can be considered as a special form of dynamic appearance where other factors (texture, illumination, etc.) are already factored out. If we consider the human silhouettes through the walking cycle as points in a high dimensional visual input space, then, given the spatial and the temporal constraints, it is expected that these points will lay on a low dimensional manifold. Intuitively, the gait is a one-dimensional manifold which is embedded in a high dimensional visual space. This was also shown in [17]. Such manifold can be twisted- and self-intersect in such a high dimensional visual space. Similarly, if we consider other human activities such as gesturing, most of the gestures are also one-dimensional manifolds. One question we aim to answer is: what is the geometric structure and properties of this manifold?

Can we decompose the configuration using linear models? Linear models, such as PCA [61], have been widely used in appearance modeling to discover subspaces for appearance variations, for example, extensively for face recognition such as in [133, 6, 148, 82, 115] and to model the appearance manifold and view manifold for 3D object recognition as in [95, 96, 49]. Such subspace analysis can be further extended to decompose other factors using bilinear models and multi-linear tensor analysis [128, 138]. In most of these cases, the object is stationary (rigid) or the motion is local (as in facial expressions).

In our case, the object is dynamic. So, can we decompose the configuration from the shape

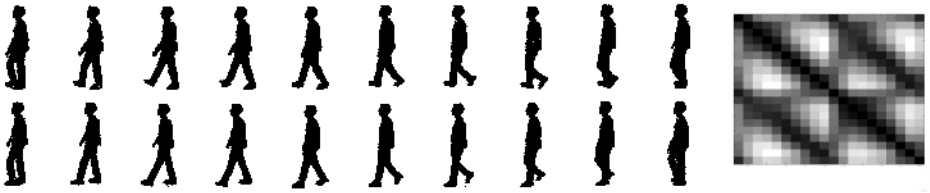


Figure 1.1: Twenty sample frames from a walking cycle from a side view. Each row represents half a cycle. Notice the similarity between the two half cycles. The right part shows the similarity matrix: each row and column corresponds to one sample. Darker means closer distance and brighter means larger distances. The two dark lines parallel to the diagonal show the similarity between the two half cycles

(appearance) using linear embedding? For our case, the shape temporally undergoes deformations and self-occlusion which result in the points lying on a nonlinear, twisted manifold. This can be illustrated if we consider the walking cycle in Fig. 1.1. The two shapes in the middle of the two rows correspond to the farthest points in the walking cycle kinematically and are supposedly the farthest points on the manifold in terms of the geodesic 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. 1.1. Because of such nonlinearity, global linear model, PCA, will not be able to discover the underlying manifold. For the same reason, multidimensional scaling (MDS) [31] also fails to recover such manifold.

1.2.2 Our Goals

In spite of significant progress in human motion analysis, tracking and synthesis, there are several limitations in the current approaches. The goals of our research for modeling of human motion are as follows:

Modeling the intrinsic structure of motion: Embedding manifolds to low dimensional Euclidian spaces provides a way to explicitly model their intrinsic structure. Learning motion manifolds can be achieved through linear subspace approximation (PCA) as in [38]. PCA have been widely used in appearance modeling to discover subspaces for appearance variations and modeling view manifolds as in [95, 133, 6, 30]. 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 in the intrinsic dimensionality of the manifold. If we know the intrinsic structure of the dynamic shape and appearance in human motion, we can utilize such intrinsic structure to achieve representations that can be much lower dimensionality. Recent advances in nonlinear manifold learning and dimensionality reduction shows the potential to find the intrinsic structure of nonlinear manifolds to achieve representations in low dimensional spaces.

Human motion is continuous movements of body joints; however, observations are discrete when we capture motions using a digital camera. Many approaches modeling dynamic sequence of human motion like hidden Markov models (HMM) describe body configuration by discrete key poses or linear combination of selected key poses. In contrast, a continuous representation of body configuration is favorable since we can estimate intermediate body configurations more accurately and synthesize motions which preserve nonlinear characteristics.

Generative model for dynamic shape and appearance without a 3D model: Accurate synthesis of facial expressions and human motions is required in computer animations for digital entertainment like films. Analysis and tracking of human motion in images also requires the ability to synthesize shape and appearance (as in Bayesian tracking). Therefore, we need to build accurate generative models for the dynamic shape and appearance.

Factorization of multiple components: The shape and appearance of human motion varies with different people, with other observation conditions such as view point and lighting. The human perceptual system routinely separates the *content* and *style* factors of their observations and the decomposition can be used for visual image analysis [128]. However, previous work focuses on static images such as several discrete static poses and person styles, and facial expression analysis in peak expression images [138, 140]. How can we extend this *style* and *content* decomposition to dynamic human motion? Where the content is the intrinsic motion and style is the way the motion is observed such as variations in shape and appearance.

Modeling two continuous manifolds (view and posture manifold): We consider tracking and inferring view and body configuration of human motion from a single monocular camera where the person can change his/her view with respect to the camera while being tracked

> (or equivalently the camera can be moving). Modeling both the view and body configuration manifolds for human motion jointly in the visual space is a very challenging task and is useful for tracking, posture estimation, and view estimation. The observation for a given body posture lies on a one-dimensional manifold (view manifold) in the visual input space. Obviously, each body posture will have it's own view manifold. If we consider a sequence of postures, making up a motion, the resulting visual manifold will become complicated as it becomes a product of the motion manifold and the view manifold. Therefore, we aim to develop a framework for modeling data laying on a two dimensional manifold (e.g. posture × view).

There are mainly three tasks we consider here using our proposed model:

Inferring body configuration and view from observation: Given a shape instance, we need to recover the body configuration and view using the learned model. This is a harder problem. Solving such problems is essential to initialize a tracker. Since the proposed model is generative, this task involves searching the model parameter space for optimal parameter that explain the data, i.e., minimize some reconstruction error. Since this involves a search over the parameter space, it is desired that the parameter space be as low dimensional and well constrained as possible.

Tracking: The proposed model can be used as a state representation. Since, in a Bayesian tracking setting, a model of the state posterior given the observation is maintained using sampling methods, it is desired that the state space be as low dimensional as possible for effective sampling. Since transition between frames is expected to be smooth both in terms of body configuration and view variations, the model provides a well behaved, dynamic model for tracking. In summation, the proposed model provides in a direct way: a low dimensional state representation, a constrained dynamic model, and an observation model.

Synthesis: The proposed model is generative and, therefore, can be used to synthesize shapes and appearances at different configurations and at different view points, and for different people, without 3D body model.

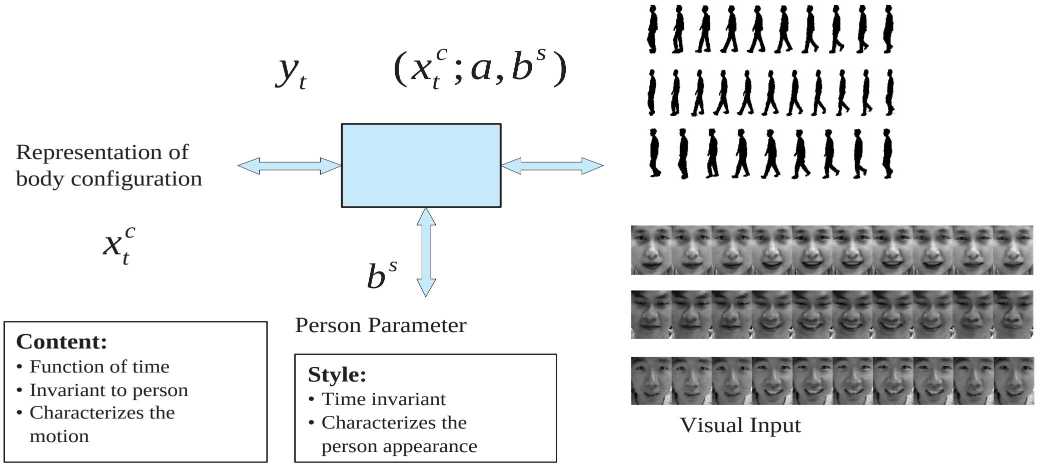


Figure 1.2: Style and content factors: Content: gait motion or facial expression. Style: different silhouette shapes or face appearance.

1.2.3 Factorized Nonlinear Generative Models Using Manifold Learning

We propose *manifold-based factorized nonlinear generative models* that support our goals for modeling dynamic shape and appearance in articulated human motions; the proposed model represents intrinsic structure in continuous low dimensional embedding space with decomposition of multiple factors affecting the observation.

Although the intrinsic body configuration manifolds might be very low in dimensionality, the resulting appearance manifold is challenging to model, given various aspects that affect the appearance. Examples of such aspects include the shape and appearance of the person performing the motion, or variation in the view point, or illumination. Such variability makes the task of learning visual manifold very challenging because we are dealing with data points that lies on multiple manifolds on the same time: body configuration manifold, view manifold, shape manifold, illumination manifold, etc.

The question we address is how to separate the style and content on a manifold representing a dynamic object. To illustrate our point we consider the human silhouette through the walking cycle (gait), such as shown in Fig. 1.2. For example, given several sequences of walking silhouettes, as in Fig. 1.2, with different people walking, how can the intrinsic body configuration be decomposed through the action *(content)* from the appearance (or shape) of the person performing the action *(style)*? In other words, given such sequences, how can we learn a generative model that explicitly factorizes the intrinsic body configuration, as a function of time that is invariant to style variations, from the personalized style of the person performing the action as a time-invariant parameter.

Manifold Embedding

In order to model intrinsic structure of dynamic shape and appearance, we utilize low dimensional embedding from the collection of shape and appearance sequence data. Unsupervised nonlinear manifold learning techniques, such as locally linear embedding (LLE) [111] and Isomap [127], are used to find intrinsic low dimensional representation of dynamic shapes and appearances. We use multiple manifold embedding to achieve decomposition of style parameters in the space of nonlinear mapping functions learned between a unified representation of the embedded manifold and visual input in style variations.

When multiple variant factors exist on the data set, data-driven manifold embedding will be quite different. These variations pose a challenge if we would like to use motion manifolds as constraints for the motion, for example, in tracking or for body pose recovery. But, conceptually, all these manifolds are the same. They are all topologically equivalent, i.e., homeomorphic to each other and we can establish a bijection between any pair of them. So, given conceptual knowledge about the topology of the manifold, we can use such knowledge in modeling real motion manifolds with different sources of variability such as different people, different views, etc. Therefore, we propose and investigate supervised manifold learning to utilize known manifold structure or idealistic topological structure.

We further investigate the role of different manifolds, such as input visual manifold, and output kinematic manifold, for embedding articulated human motion in low dimensional space. Most manifold-based approaches from visual data are limited so far to simple kinds of motion such as walking, simple gestures, or golf swings, which are mainly one dimensional in nature. For approaches which aim to model the visual manifold, as in [36, 23, 92], there is another fundamental limitation of being view-based. The problem stems from the complexity of the visual manifolds, if continuous view variations are considered. We introduce a model that ties together the body configuration (kinematics) manifold and the visual manifold (observations) in a way that facilitates tracking the 3D configuration with continued relative view variability. The model exploits the low dimensionality nature of both the body configuration manifold and the view manifold where each of them are represented separately.

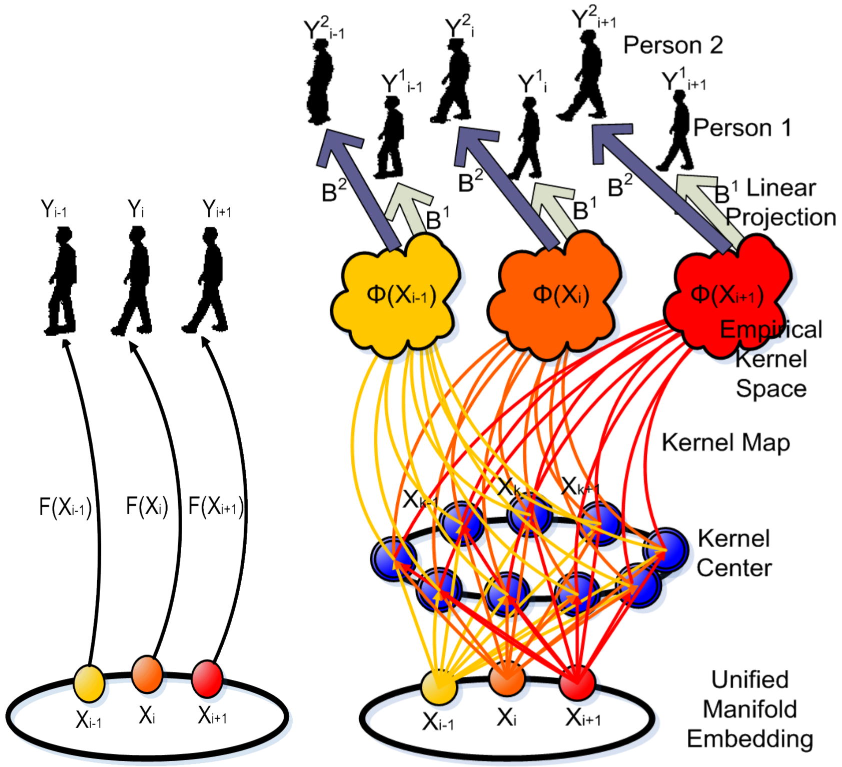


Figure 1.3: Nonlinear Mapping from Embedding Manifold

Nonlinear Mapping

In any motion sequence, there is a corresponding manifold point for every input shape and/or appearance. The relation between the intrinsic configuration state and the observed shape and/or appearance is nonlinear, since we utilize nonlinear manifold embedding. Therefore, we learn nonlinear mapping between the embedding space and the observations. Such mapping facilitates the generation of the original motion sequence accurately from the embedded representation. Fig. 1.3 shows nonlinear mapping from one-dimensional circular manifold embedding to the observation silhouette. For each sequence of motion, we have unique nonlinear mapping that can generate the original sequence of motion in any intermediate points from a continuous manifold embedding.

Any nonlinear mapping that minimizes the regularized risk can be represented in the form [70]:

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}), \tag{1.1}$$

given a set of patterns $\{\boldsymbol{x}_i\}$, empirical kernel $k(\boldsymbol{x}_i, \boldsymbol{x})$, and target values $\boldsymbol{y}_i = f(\boldsymbol{x}_i)$. The solutions lie on the linear span of kernels centered on data points, $\boldsymbol{x}_i$'s. That is, any nonlinear mapping is equivalent to a linear projection from a kernel map space. In our case, any nonlinear mapping from the embedding space to the dynamic shape and appearance can be represented by a kernel map from the embedding manifold and a linear projection from the kernel space. Given a unified manifold representation of different sequences of motion, we can have a unique kernel map. This kernel map allows modeling of each motion sequence of a different number of frames with the same form of linear projection from the kernel space.

Factorization

Given several sequences of dynamic shapes from different people, the nonlinear mapping for each person's sequence will be different due to shape style variations in different sequence. By factorizing this nonlinear mapping, we can represent the dynamic shape and appearance using decomposable generative models. We utilize multilinear analysis to decompose the nonlinear mapping coefficients of the dynamic shape and appearance mapping into orthogonal factors. Multilinear analysis can be achieved by higher-order singular value decomposition (HOSVD), which is a generalization of SVD [72][139]. Multilinear analysis for the linear projection from the same kernel map provides factorized nonlinear generative models for the observed dynamic motions.

1.3 Contributions

Contributions of this dissertation are summarized in the following;

Generative Models for Dynamic Shape and Appearance Models Using Manifold Embedding

We introduce a framework that aims to learn a landmark-free, correspondence-free global representation of dynamic appearance manifolds. The framework is based on using nonlinear dimensionality reduction to achieve an embedding of the global deformation manifold that preserves the geometric structure of the manifold. Given such embedding, a nonlinear mapping is learned from such embedded space into visual input space. We use Radial Basis Function (RBF) interpolation framework for such nonlinear mapping. Therefore, any visual input is represented by a linear combination of nonlinear bases functions centered along the manifold in the embedded space. We also show how approximate solution for the inverse mapping can be obtained in a closed form, which facilitates the recovery of the intrinsic body configuration. We use the framework to learn a representation of the gait manifold as an example of a dynamic shape manifold and show how the learned representation can be used to interpolate intermediate body poses, as well as in recovery and reconstruction of the input. As a direct application of learning the gait manifold, we present a framework for recovery of 3D body pose and view point from silhouettes. We also show examples of using the framework in learning the manifolds for some simple gestures and facial expressions as examples of dynamic appearance manifolds.

New Framework for Separating Style and Content on Nonlinear Manifolds

We introduce a novel framework for separating style and content on manifolds representing dynamic objects. We learn a factorized generative model that explicitly decomposes the intrinsic body configuration (content) as a function of time from the appearance (style) factor(s) of the person performing the action as time-invariant parameters. The framework is based on decomposing the style parameters in the space of nonlinear functions that map between a learned unified nonlinear embedding of multiple content manifolds and the visual input space. The learned model supports tasks such as synthesis, body configuration recovery, recovery of other aspects such as view, person parameters, etc. As direct and important applications of the introduced framework, we consider the case of gait and also show results for facial expressions.

Decomposition of Multiple Factors

Using a manifold embedding invariant to the observation variability, we achieve decomposition of multiple factors that affect the observation. The empirical kernel space from the embedded manifold allows us to analyze multiple factors by projection matrices, where we can apply multiple linear analysis. Still, the overall generative model preserves the nonlinear characteristics of dynamic shape and appearance.

In our experiments, we learn generative models that can generate walking silhouettes for different people from different view points. Given a single image or a sequence of images, we can use the model to solve for the body configuration, view and person shape style parameters. As a result we can directly infer 3D body pose, view point, and person shape style from the visual input. We also apply the model for facial expressions as an example of a dynamic appearance. In this case we learn a generative model that can generate different dynamic facial expressions in different people. The model can successfully be used to recognize expressions performed by a new person never seen in the training.

Style Adaptive Tracking

We can find compact, low dimensional representation of body configuration for tracking by applying explicit nonlinear manifold learning and its parametric representation. Then, we achieve adaptive tracking of a person shape by estimating style parameters according to observed visual input. The adaptive style parameter estimation allows not only tracking of any new person contour but also identification of the person during tracking. As a result, we achieve robust, adaptive tracking with simultaneous style estimation from a cluttered environment.

Modeling Continuous View and Body Pose Manifold

We can deal with both body configuration and view points as continuous variables in a product space (one dimensional *view manifold* $\times$ one dimensional *body configuration manifold*) using torus manifold. This facilitates tracking subjects with varying view points due to camera motion or changing subject view with respect to the camera. In addition, we propose a framework for modeling both the configuration and view manifolds using kinematics manifold. We use kinematics manifold as a representation of the configuration invariant to view. Given the kinematic manifold, the view manifold is then explicitly modeled in the nonlinear mapping space between the kinematics manifold embedding and the view-variant observations. The result is two low-dimensional embeddings: one for configuration and one for the view. This model provides another product manifold that can generate observation given the two manifolds' state parameters. This fits perfectly into the Bayesian tracking as it provides in a direct way: 1) low dimensional state representation for both view and configuration, 2) a constrained dynamic model since the manifolds are modeled explicitly, and 3) an observation model, which comes directly from the generative model used.

Chapter 2

Related Work

2.1 Human Motion Analysis

In the last decade there has been extensive research in human motion analysis. We refer the reader to [40, 3, 90] for a comprehensive survey of the broad subject. In this chapter, we review three directly related areas: inferring 3D body pose, manifold-based tracking, and gait recognition.

2.1.1 Inferring 3D Body Pose

Recovery of 3D body pose is a fundamental problem for human motion analysis in many applications such as motion capture, vision interface, visual surveillance, and gesture recognition. The human body is an articulated object that moves through the three-dimensional world. This motion is constrained by 3D body kinematics and dynamics, as well as the dynamics of the activity being performed. Such constraints are explicitly exploited to recover the body configuration and motion in model-based approaches, such as [62, 54, 107, 106, 42, 64, 119], through explicitly specifying articulated models of the body parts, joint angles and their kinematics (or dynamics), as well as models for camera geometry and image formation. Recovering body configuration in these approaches involves searching high dimensional spaces (body configuration and geometric transformation), which is typically formulated deterministically as a non-linear optimization problem, e.g. [106], or probabilistically as a maximum likelihood problem, e.g. [119]. Such approaches achieve significant success when the search problem is constrained as in a tracking context. However, initialization remains the most challenging problem, which can be partially alleviated by sampling approaches. Partial recovery of body configuration can also be achieved through intermediate view-based representations (models) that may or may not be tied to specific body parts [33, 21, 146, 63, 115]. Alternatively, 3D body pose can be directly inferred from the visual input [55, 18, 109, 110, 94, 45, 116, 2, 123]. We call such approaches learning-based, since their objective is to directly infer the 3D body pose as a function of the visual input. Such approaches have great potential in solving the fundamental initialization problem for model-based vision.

Inferring 3D pose from silhouettes can be achieved by learning mapping functions from the visual input to the pose space. However, learning such mapping between high dimensional spaces from examples is fundamentally an ill-posed problem. Therefore, certain constraints have to be exploited. In [109, 110], learning specialized nonlinear mappings from Hu moment representation of the input shape and the pose space facilitated successful recovery of the pose directly from the visual input. In [18], the problem was constrained using nonlinear manifold learning where the pose is inferred by mapping sequences of the input to paths of the learned manifold. In [55], the reconstruction was based on 2D tracking of joints and a probabilistic model for human motion. In [45], 3D structure is inferred from multi-view using a probabilistic model of multi-view silhouettes and key points on the object. The inferring pose can also be posed as a nearest neighbor search problem where the input is matched to a database of exemplars with known 3D pose. In [94], pose is recovered by matching the shape of the silhouette using shape context. In [116], locality sensitive hashing was used to efficiently match local models from the input to large exemplar sets. However, in almost all these approaches, the mapping is learned from a representation of visual input to 3D or other intermediate representations, which is highly under-constrained mapping which can lead to poor generalization. In addition, such discriminative mapping has an inherent ambiguity problem that needs to be addressed as in [123]

2.1.2 Manifold-based Tracking

Tracking the human body and recovery of 3D body pose is a challenging problem for human motion analysis with many applications such as visual surveillance, human-machine interface, and gesture recognition. Traditionally, this problem has been addressed through generative approaches that map from a 3D body configuration space to the visual observation space, e.g., [62, 54, 107, 106, 42, 64, 119, 105]. Therefore, the recovery of the 3D configuration is formulated as a search problem for the best configuration that minimizes an error metric given the visual observation, e.g., [106, 119]. Such approaches typically require a body model and a calibrated camera in order to obtain hypothesis observations from configurations. Similarly, 2D view-based body models can be used [63, 57]; however, this is limited in dealing with continuous view variability.

Recently, researchers [18, 122, 104, 135, 92, 91, 134], including our work [36, 79], have increasing interest into constraining the problem by exploiting the fact that despite the high dimensionality of the body configuration space, many human motion activities lie intrinsically on low dimensional manifolds. This can be achieved through learning the body configuration manifold, as in [122, 135], which brings a better dynamic-modeling for tracking. Alternatively, this can be achieved through learning the visual input manifold, as in [36, 23, 92], which helps recovery of configuration from the visual input.

2.1.3 Gait Tracking and Recognition

Human gait is a valuable biometric cue that can be used for human identification similar to other biometrics, such as faces and fingerprints. Gait has significant advantages compared to other biometrics since it is easily observable in an unintrusive way and is difficult to disguise [32]. Therefore, gait recognition has a great potential for human identification in public spaces for surveillance and for security [32, 60, 112, 56]. A fundamental challenge in gait recognition is to develop robust recognition algorithms that can extract gait features that are invariant to the presence of various conditions which affect people's appearance. As a challenging problem in gait recognition, different conditions such as view, clothing, walking surface, and shoe type were presented in the NIST dataset [112]. Many gait recognition algorithms assume constrained conditions to reduce various sources that influence recognition accuracy. Two typical assumptions are fixed view (especially side view), and constant speed. It is challenging to develop a gait recognition system invariant to a different view and different speed.

The appearance of gait in image sequences is a spatiotemporal process that characterizes the walker. Gait recognition algorithms, generally, aim to capture discriminative spatiotemporal features from image sequences in order to achieve human identification. We can categorize gait-recognition approaches into model-based approaches and appearance-based approaches according to the features used for classification. Model-based approaches [32, 80, 60, 13] fit models or intermediate body representations in order to extract proper features (parameters) that describe the dynamics of the gait. Appearance-based approaches aim to capture a spatiotemporal gait characteristic directly from input sequences.

For robust recognition of gait, several different features are used. Murase [96] used parametric eigenspace representation to represent a moving object using Principle Component Analysis (PCA). Huang *et al.* [56] extended the method using Canonical space transformation (CST) based on Canonical Anaylsis (CA), with eigenspace transformation for feature extraction. BenAbdelkader *et al.* [9] used self-similarity measurements to capture spatiotemporal characteristics using PCA analysis. Hayfron-Acquah *et al.* [52] used symmetric information to capture gait motion. Little *et al.* [85] computed phase vector from extracted optical flow. Shutler *et al.* [118] used higher order moments. In [53], HMM was used to capture gait dynamics from quantized Hu moments of silhouettes. HMM was also used in [65] with features representing silhouette width distribution. Still, it was difficult to extract good features to capture gait characteristics of individual people. The proposed approach based on bilinear and multilinear analysis of gait in different people after temporal normalization provides a new good feature to distinguish individuals for gait recognition. Extracting a new gait feature that is invariant to temporal variations and other factors is challenging.

2.2 Representation of Dynamic Shape and Appearance

2.2.1 Linear Models

Applying linear models are landmark-based approaches where correspondences are established between these landmarks. Examples of such approaches include active shape models and active appearance models [30], where deformations in the shape and appearance are modeled through linear models of certain landmarks through a correspondence frame. A fundamental problem is that such correspondences are not always feasible (has no meaning). For example, if there are changes in the topology over time (as in our gait example), correspondences between landmarks are not always feasible because of self occlusion and self similarity. For these reasons, correspondence-free vector representations (global) have the advantage of implicitly imposing a correspondence frame, as well as not requiring explicit landmarks (feature extraction) to be identified. Therefore, vectorial representations have been attractive in modeling appearance such as in [133, 95, 14]. Recovering global geometric transformations for such appearance representations has been addressed in [14, 39]. In case of dynamic shape and appearance like gait, the class of deformation cannot be modeled using such global geometric transformations.

So, how can global, landmark-free, correspondence-free vectorial representation be used for dynamic objects where, obviously, the implicit correspondences between individual vector components do not hold because of the motion? We argue that explicit modeling of the manifold will make this possible. Although, globally (in time) the implicit correspondences enforced by the vectorial representation do not hold, locally (along the manifold) such implicit correspondences are quite valid between each point and its manifold neighbors. Because of the nonlinearity of the dynamic shape and appearance manifolds, we need to use a framework that is able to recover the underlying nonlinear manifold.

2.2.2 Learning Visual Manifolds

Learning nonlinear deformation manifolds is typically performed in the visual input space or through intermediate representations. Learning motion manifolds can be achieved through linear subspace approximation, as in [38, 43]. Alternatively, exemplar-based approaches such as [131, 39] implicitly model nonlinear manifolds through points (exemplars) along the manifold. Such exemplars are represented in the visual input space. HMM models provide a probabilistic piecewise linear approximation of the manifold, which can be used to learn nonlinear manifolds as in [20] and in [18].

Recently, increasing research interest has focused on explicitly modeling motion manifolds and exploring how that can be useful in constraining the task of tracking or recovering body configurations. In our work [36], which is a part of this dissertation, the visual manifolds of human silhouette deformations, due to motion, have been learned explicitly and used for recovering 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. Simultaneously in [122], manifold learned from the body configuration space is used to provide constraints for tracking. Later, in [136, 134], learning the body configuration manifold provided a way to learn nonlinear dynamic models through Gaussian processes which constrains the tracking. In [91], models that coupled learning dynamics with embedding were introduced. It was also shown in [93] that learning the motion manifolds provides ways to establish correspondences between subjects observed from different cameras. In contrast to learning the motion manifolds, as in [36, 122, 136], learning the shape manifold, as in [129], provides a way to constrain the recovery of body pose from visual input.

Deformation in shape has been studied in various scientific disciplines. In computer vision, both variational approaches [68, 99, 100, 124] and statistical approaches are used to model shape deformations. Statistical approaches model shape deformations as statistical variations within the shape population [69, 16, 30, 81]. Modeling shape deformation is a key issue for several related problems such as shape matching, shape classification, contour tracking, and image segmentation. Landmark-free deformable templates have been introduced in [47, 147]

2.3 Factorized Models

Subspace analysis by linear model such as PCA can be extended to decompose multiple orthogonal factors using bilinear models and multilinear tensor analysis [128, 138]. The pioneering work of Tenenbaum and Freeman [128] formulated the separation of style and content using a bilinear model framework [87]. In this 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 [87, 88]. In [138], 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. They employed n-mode SVD [72] to fit multilinear models. Tensor representation of image data was used in [117] for video compression and in [137] for motion analysis and synthesis. N-mode analysis of higher-order tensors was originally proposed and developed in [132, 67, 87] and others. See details of multilinear analysis in Appendix A. The applications of bilinear and multilinear models, as in [128, 138], to decompose variations into orthogonal factors are mainly for static image ensembles.

2.3.1 Limitations of Bilinear and Multilinear Factorization

The question we address 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 if we consider the walking cycle example in Fig. 1.1. In this case, the shape temporally undergoes deformations and self-occlusion, which result in the points lying on a nonlinear, twisted manifold. The two shapes in the middle of the two rows correspond to the farthest points in the walking cycle kinematically and are supposedly the farthest points on the manifold in terms of the geodesic 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. 1.1. Because of such nonlinearity, PCA, bilinear, multilinear models will not be able to discover the underlying manifold and decompose orthogonal factors.

Another limitation of bilinear and multilinear analysis, as presented in [128, 138], is that it is mainly a supervised procedure where the image ensemble needs to be arranged into various style, content or orthogonal factor classes beforehand. Such requirement makes it hard if we try to use bilinear or multilinear models with image sequences to decompose orthogonal factors on a manifold. Typically, input sequences can be of different lengths, with different sampling rates, and with people performing the same activity with different dynamics. So we aim to have an unsupervised procedure with minimal human interaction.

2.4 Manifold Learning

Embedding nonlinear manifolds to low dimensional Euclidian spaces provides a way to explicitly model such manifolds. Learning motion manifolds can be achieved through linear subspace approximation (PCA), as in [38]. PCA have been widely used in appearance modeling to discover subspaces for appearance variations and modeling view manifolds, as in [95, 133, 6, 30]. 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 in the intrinsic dimensionality of the manifold.

Recently some promising frameworks for nonlinear dimensionality reduction have been introduced including isometric feature mapping (Isomap) [126] and locally linear embedding (LLE) [111]. Related nonlinear dimensionality reduction work also includes [19, 7, 74, 145]. Such approaches can achieve embedding of nonlinear manifolds through changing the metric from the original space to the embedding space based on local structure of the manifold. While there are various such approaches, they mainly fall into two categories: Spectral-embedding approaches and Statistical approaches. Spectral embedding includes approaches such as isometric feature mapping (Isomap) [126], Local linear embedding (LLE) [111], Laplacian eigenmaps [7], and Manifold Charting [19]. Spectral-embedding approaches in general construct an affinity matrix between data points that reflects local manifold structure. Embedding is then achieved through solving an eigen-value problem on such matrix. It was shown in [11, 51] that these approaches are all instances of kernel-based learning, in particular kernel principle component analysis KPCA[113]. In [10], an approach for embedding out-of-sample points is proposed to complement such approaches. Along the same line, our work introduces a general framework for mapping between input and embedding spaces and to factorize style factors in this mapping space.

Nonlinear dimensionality reduction approaches are able to embed image ensembles nonlinearly into low dimensional spaces where various orthogonal perceptual aspects can be shown to correspond to certain directions or clusters in the embedding spaces. In this sense, such nonlinear dimensionality reduction frameworks 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 or how to model multiple manifolds.

Chapter 3

Nonlinear Manifold Learning for Dynamic Shape and Dynamic Appearance

We introduce a framework that aim to learn a landmark-free correspondence-free global representations of dynamic appearance manifolds. We use nonlinear dimensionality reduction to achieve an embedding of the global deformation manifold that preserves the geometric structure of the manifold. Given such embedding, a nonlinear mapping is learned from the embedding space into the visual input space. Therefore, any visual input is represented by a linear combination of nonlinear bases functions centered along the manifold in the embedding space. We also show how approximate solution for the inverse mapping can be obtained in a closed form which facilitate recovery of the intrinsic body configuration. We use the framework to learn the gait manifold as an example of a dynamic shape manifold, as well as to learn the manifolds for some simple gestures and facial expressions as examples of dynamic appearance manifolds.

3.1 Motivation

Our objectives is to learn representations for the shape and the appearance of moving (dynamic) objects that supports tasks such as synthesis, pose recovery, reconstruction and tracking. Such learned representation will serve as view-based generative models for dynamic appearance in the form

$$\boldsymbol{y}_t = T_{\boldsymbol{\alpha}} \gamma(\boldsymbol{x}_t; \boldsymbol{a}) \tag{3.1}$$

where the appearance, $\boldsymbol{y}_t$, at time t is an instance driven from a generative model where the function γ is a mapping function that maps body configuration $\boldsymbol{x}_t$ at time t into the image space. i.e., the mapping function γ maps from a representation of the body configuration space into the image space given mapping parameters $\boldsymbol{a}$ that are independent from the configuration. $T_{\boldsymbol{\alpha}}$ represents a global geometric transformation on the appearance instance.

3.2 Embedding Nonlinear Manifolds

3.2.1 Representation

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 such that

$$y(x) = \begin{cases} d_c(x) & x \text{ inside } c \\ 0 & x \text{ on } c \\ -d_c(x) & x \text{ outside } c \end{cases}$$

where the $d_c(x)$ is the distance to the closest point on the contour c with a positive sign inside the contour and a negative sign outside the contour. Such representation impose smoothness on the distance between shapes. Given such representation, the input shapes are points $boldsymboly_i \in \mathbb{R}^d, i = 1, \cdots, N$ where d is the same as the dimensionality of the input space and N is the number of points. Implicit function representation is typically used in level-set methods.

Appearance Representation

Appearance is represented directly in a vector form, i.e., each instance of appearance is represented as points $\boldsymbol{y}_i \in \mathbb{R}^d, i = 1, \cdots, N$ where d is the dimensionality of the input space.

3.2.2 Embedding

Because of the nonlinearity of the dynamic shape and appearance manifolds we need to use a framework that is able to recover the underlying nonlinear manifold. We adapt the LLE framework [111]. Given the assumption that each data point and its neighbors lie on a locally linear patch of the manifold [111], each point (shape or appearance instance) y_i can be reconstructed based on a linear mapping $\sum_j w_{ij} y_j$ that weights its neighbors contributions using the weights w_{ij}. In our case, the neighborhood of each point is determined by its K nearest neighbors based on the distance in the input space. The objective is to find such weights that minimize the global reconstruction error,

$$E(W) = \sum_i |\boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_j|^2 \qquad\qquad i, j = 1 \cdots N. \tag{3.2}$$

The weights are constrained such that w_{ij} is set to 0 if point $\boldsymbol{y}_j$ is not within the K nearest neighbors of point $\boldsymbol{y}_i$. This will guarantee that each point is reconstructed from its neighbors only. The weights obtained by minimizing the error in Eq. 3.2 are invariant to rotations and re-scalings. To make them invariant to translation, the weights are also constrained to sum up to one across each row, i.e., the minimization is subject to $\sum_j w_{ij} = 1$. Such symmetric properties are essential to discover the intrinsic geometry of the manifold independent of any frame of reference. Optimal solution for such optimization problem can be found by solving a least-squares problem as was shown in [111].

Since the recovered weights $\boldsymbol{W}$ reflects the intrinsic geometric structure of the manifold, an embedded manifold in a low dimensional space can be constructed using the same weights. This can be achieved by solving for a set of points $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^e, i = 1 \cdots N\}$ in a low dimension space, $e \ll d$, that minimizes

$$E(\boldsymbol{X}) = \sum_i |\boldsymbol{x}_i - \sum_j w_{ij} \boldsymbol{x}_j|^2 \qquad i, j = 1 \cdots N, \tag{3.3}$$

where in this case the weights are fixed. Solving such problem can be achieved by solving an eigenvector problem as was shown in [111].

One point that need to be emphasized is that we do not use the temporal relation to achieve the embedding, since the goal is to obtain an embedding which preserves the geometry of the manifold. Temporal relation can be used to determine the neighborhood of each shape but that would lead to erroneous embedding if there is no enough samples on the manifold.

3.3 Nonlinear Mapping: Learning Generative Model

Given a visual input, the objective is to recover the intrinsic body configuration by finding the point on the manifold in the embedding space corresponding to this input. Recovering such embedded representation will facilitate reconstruction of the input and detection of any spatial or temporal outliers. In other words, we aim to simultaneously solve for the pose and reconstruct the input. To achieve this goal, two steps are required:

1. we need to model the appearance manifold given the Euclidean space embedding achieved in the previous section.

2. we need to learn a mapping between the embedding space and the visual input space.

The manifold in the embedding space can be modeled explicitly in a function form or implicitly by points along the embedded manifold (embedded exemplars). The embedded manifold can be also modeled probabilistically using Hidden Markov Models and EM. Clearly, learning manifold representations in a low-dimensional embedding space is advantageous over learning them in the visual input space. However, our emphasize is on learning the mapping between the embedding space and the visual input space.

Since the objective is to recover body configuration from the input, it might be obvious that we need to learn mapping from the input space to the embedding space, i.e., mapping from $\mathbb{R}^d$ to $\mathbb{R}^e$. However, learning such mapping is not feasible since the visual input is very high-dimensional so learning such mapping will require large number of samples in order to be able to interpolate. Instead, we learn the mapping from the embedding space to the visual input space with a mechanism to directly solve for the inverse mapping.

It is well know that learning a smooth mapping from examples is an ill-posed problem unless the mapping is constrained since the mapping will be undefined in other parts of the space [103]. We Argue that, explicit modeling of the visual manifold represents a way to constrain any mapping between the visual input and any other space. Nonlinear embedding of the manifold, as was discussed in the previous section, represents a general framework to achieve this task. Constraining the mapping to the manifold is essential if we consider the existence of outliers (spatial and/or temporal) in the input space. This also facilitates learning mappings that can be used for interpolation between poses as we shall show. In what follows we explain our framework to recover the pose. In order to learn such nonlinear mapping we use Radial basis function (RBF) interpolation framework. The use of RBF for image synthesis and analysis has been pioneered by [103, 12] where RBF networks were used to learn nonlinear mappings between image space and a supervised parameter space. In our work we use RBF interpolation framework in a novel way to learn mapping from unsupervised learned parameter space to the input space. Radial basis functions interpolation provides a framework for both implicitly modeling the embedded manifold as well as learning a mapping between the embedding space and the visual input space. In this case, the manifold is represented in the embedding space implicitly by selecting a set of representative points along the manifold.

Let the set of representative input instances (shape or appearance) be $\mathsf{Y} = \{\boldsymbol{y}_i \in \mathbb{R}^d \quad i = 1, \cdots, N\}$ and let their corresponding points in the embedding space be $\mathsf{X} = \{\boldsymbol{x}_i \in \mathbb{R}^e, \quad i = 1, \cdots, N\}$ where e is the dimensionality of the embedding space (e.g. $e = 3$ in the case of gait). We can solve for multiple interpolants $f^k : \mathbb{R}^e \to R$ where k is k-th dimension (pixel) in the input space and f^k is a radial basis function interpolant, i.e., we learn nonlinear mappings from the embedding space to each individual pixel in the input space. Of particular interest are functions of the form

$$f^k(\boldsymbol{x}) = p^k(\boldsymbol{x}) + \sum_{i=1}^{N} w_i^k \phi(|\boldsymbol{x} - \boldsymbol{x}_i|), \tag{3.4}$$

where $\phi(\cdot)$ is a real-valued basic function, w_i are real coefficients, $|\cdot|$ is the norm on $\mathbb{R}^e$ (the embedding space). Typical choices for the basis function includes thin-plate spline ($\phi(u) = u^2 log(u)$), the multiquadric ($\phi(u) = \sqrt{(}u^2 + c^2)$), Gaussian ($\phi(u) = e^{-cu^2}$), biharmonic ($\phi(u) = u$) and triharmonic ($\phi(u) = u^3$) splines. p^k is a linear polynomial with coefficients c^k, i.e., $p^k(\boldsymbol{x}) = [1 \quad \boldsymbol{x}^\top] \cdot \boldsymbol{c}^k$. This linear polynomial is essential to achieve approximate solution for the inverse mapping as will be shown.

The whole mapping can be written in a matrix form as

$$f(\boldsymbol{x}) = \boldsymbol{B} \cdot \psi(\boldsymbol{x}), \tag{3.5}$$

where $\boldsymbol{B}$ is a $d \times (N+e+1)$ dimensional matrix with the k-th row $[w_1^k \cdots w_N^k \quad c^{k^T}]$ and the vector $\psi(\boldsymbol{x})$ is $[\phi(|\boldsymbol{x} - \boldsymbol{x}_1|) \cdots \phi(|\boldsymbol{x} - \boldsymbol{x}_N|) \;\; 1 \;\; \boldsymbol{x}^\top]^\top$. The matrix $\boldsymbol{B}$ represents the coefficients for d different nonlinear mappings, each from a low-dimension embedding space into real numbers.

To insure orthogonality and to make the problem well posed, the following additional constraints are imposed

$$\sum_{i=1}^{N} w_i p_j(x_i) = 0, j = 1, \cdots, m \tag{3.6}$$

where p_j are the linear basis of p. Therefore the solution for B can be obtained by directly solving the linear systems

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{P} \\ \boldsymbol{P}^\top & \boldsymbol{0} \end{pmatrix} \boldsymbol{B}^\top = \begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{0}_{(e+1)\times d} \end{pmatrix}, \tag{3.7}$$

where $\boldsymbol{A}_{ij} = \phi(|\boldsymbol{x}_j - \boldsymbol{x}_i|), \quad i,j = 1\cdots N$, $\boldsymbol{P}$ is a matrix with *i*-th row $[1 \quad \boldsymbol{x}_i^\top]$, and $\boldsymbol{Y}$ is $(N \times d)$ matrix containing the representative input images, i.e., $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_N]^\top$. Solution for $\boldsymbol{B}$ is guaranteed under certain conditions on the basic functions used.

Similarly, mapping can be learned using arbitrary centers in the embedding space (not necessarily at data points) [103]. In this case, given N_t centers $\{\boldsymbol{t}_j \in \mathbb{R}^e, j = 1, \cdots, N_t\}$ and given a set input images $\mathsf{Y} = \{\boldsymbol{y}_i, i = 1, \cdots, N\}$ where their corresponding embedding are $\mathsf{X} = \{\boldsymbol{x}_i, i = 1, \cdots, N\}$, we can learn interpolants in the form

$$f^k(\boldsymbol{x}) = p^k(\boldsymbol{x}) + \sum_{i=1}^{N_t} w_i^k \phi(|\boldsymbol{x} - \boldsymbol{t}_i|), \tag{3.8}$$

that satisfies the interpolation condition

$$y_i^k = f^k(\boldsymbol{x}_i) \tag{3.9}$$

which yields a system of equation

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{P}_x \\ \boldsymbol{P}_t^\top & \boldsymbol{0} \end{pmatrix} \boldsymbol{B}^\top = \begin{pmatrix} \boldsymbol{Y} \\ 0_{(e+1)\times d} \end{pmatrix}, \tag{3.10}$$

where $\boldsymbol{A}$ is $N \times N_t$ matrix with $\boldsymbol{A}_{ij} = \phi(|\boldsymbol{x}_i - \boldsymbol{t}_j|), \quad i = 1\cdots N, j = 1\cdots N_t$, $\boldsymbol{P}_x$ is a $N \times (e+1)$ matrix with *i*-th row $[1 \ \ \boldsymbol{x}_i^\top]$, $\boldsymbol{P}_t$ is a $N_t \times (e+1)$ matrix with *i*-th row $[1 \ \ \boldsymbol{t}_i^\top]$.

Given such mapping, any input is represented by a linear combination of nonlinear functions centered in the embedding space along the manifold. Equivalently, this can be interpreted as a form of basis images (coefficients) that are combined nonlinearly using kernel functions centered along the embedded manifold.

3.3.1 Solving For the Embedding Coordinates

Given a new input $\boldsymbol{y} \in \mathbb{R}^d$, it is required to find the corresponding embedding coordinates $\boldsymbol{x} \in \mathbb{R}^e$ by solving for the inverse mapping. There are two questions that we might need to answer

1. What is the coordinates of point $\boldsymbol{x} \in \mathbb{R}^e$ in the embedding space corresponding to such input.

2. What is the closest point on the embedded manifold corresponding to such input.

In both cases we need to obtain a solution for

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \, ||\boldsymbol{y} - \boldsymbol{B}\psi(\boldsymbol{x})|| \tag{3.11}$$

where for the second question the answer is constrained to be on the embedded manifold. In the cases where the manifold is only one dimensional, (for example in the gait case, as will be shown) only one dimensional search is sufficient to recover the manifold point closest to the input. However, we show here how to obtain a closed-form solution for $\boldsymbol{x}^*$.

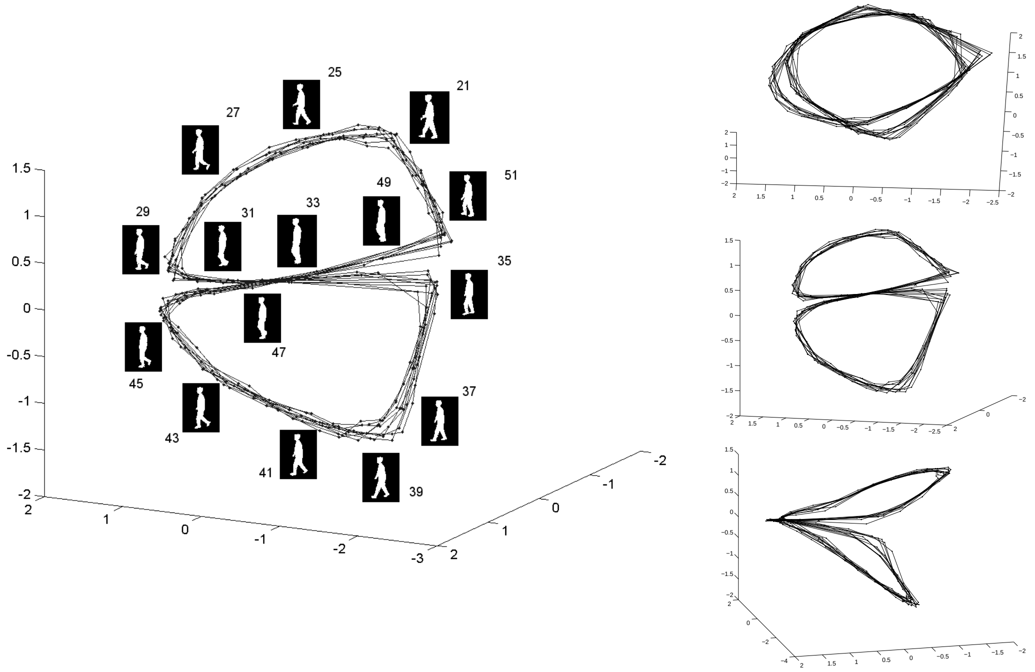


Figure 3.1: Embedded gait manifold for a side view of the walker. Left: sample frames from a walking cycle along the manifold with the frame numbers shown to indicate the order. Ten walking cycles are shown. Right: three different views of the manifold.

Each input yields a set of d nonlinear equations in e unknowns (or d nonlinear equations in one e-dimensional unknown). Therefore a solution for $\boldsymbol{x}^*$ can be obtained by least square solution for the over-constrained nonlinear system in 3.11. However, because of the linear polynomial part in the interpolation function, the vector $\psi(\boldsymbol{x})$ has a special form that facilitates a closed-form least square linear approximation and therefore, avoid solving the nonlinear system. This can be achieved by obtaining the pseudo-inverse of $\boldsymbol{B}$. Note that $\boldsymbol{B}$ has rank N since N distinctive RBF centers are used. Therefore, the pseudo-inverse can be obtained by decomposing $\boldsymbol{B}$ using SVD such that $\boldsymbol{B} = \boldsymbol{USV}^\top$ and, therefore, vector $\psi(\boldsymbol{x})$ can be recovered simply as

$$\psi(\boldsymbol{x}) = \boldsymbol{V}\tilde{\boldsymbol{S}}\boldsymbol{U}^T\boldsymbol{y} \tag{3.12}$$

where $\tilde{\boldsymbol{S}}$ is the diagonal matrix obtained by taking the inverse of the nonzero singular values in $\boldsymbol{S}$ the diagonal matrix and setting the rest to zeros. Linear approximation for the embedding coordinate $\boldsymbol{x}$ can be obtained by taking the last e rows in the recovered vector $\psi(\boldsymbol{x})$. Reconstruction can be achieved by re-mapping the projected point.

3.4 Learning the gait manifold

3.4.1 Embedding

In this section we show an example of learning the nonlinear manifold of the gait as an example of a dynamic shape. We used data sets of walking people from multiple views. Each data set consists of 300 frames and each containing about 8 to 11 walking cycles of the same person from a certain view points[1]. We applied the LLE frameworks to discover the geometric structure of the gait manifold as well as to establish a low dimensional embedding of such manifold. We also applied Isomap [127] framework on the same data to validate the results. Both Isomap and LLE resulted in qualitatively similar manifold embedding.

As a result of nonlinear dimensionality reduction we can reach an embedding of the gait manifold in a low dimension Euclidean space. Fig. 3.1 illustrates the resulting embedded manifold for a side view of the walker. Fig. 3.2 illustrates the embedded manifolds for five different view points of the walker. For a given view point, the walking cycle evolves along a closed curve in the embedded space, i.e., only one degree of freedom controls the walking cycle which corresponds to the constrained body pose as a function of the time. Such conclusion is conforming with the intuition that the gait manifold is one dimensional.

[1] The data used are from the CMU Mobo gait data set which contains 25 people from six different view points. The walkers were using treadmill which might results in different dynamics from the natural walking.

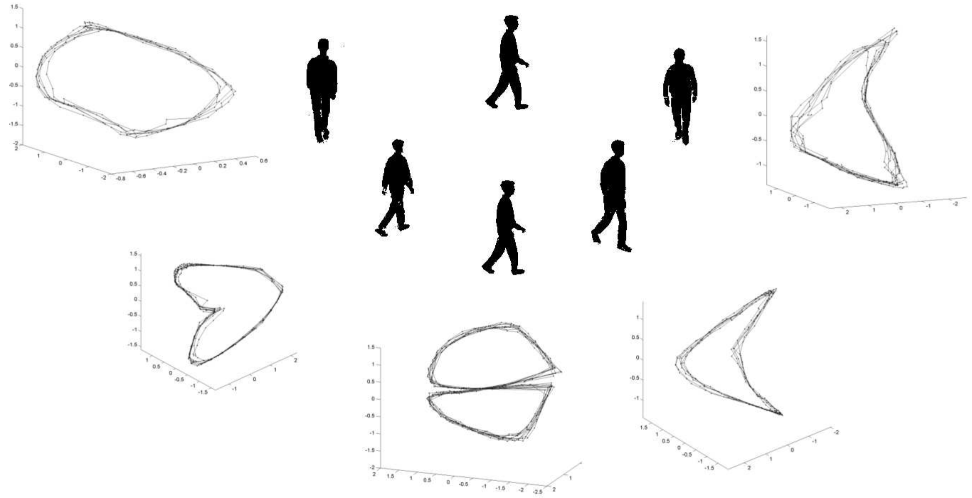


Figure 3.2: Embedded manifolds for 5 different views of the walkers. Frontal view manifold is the right most one and back view manifold is the leftmost one. We choose the view of the manifold that best illustrates its shape in the 3D embedding space

The question is what is the least dimensional embedding space we can use to embed the walking cycle in a way that discriminate different poses through the whole cycle. The answer depends on the view point. The manifold twists in the embedding space given the different view points which impose different self occlusions. The least twisted manifold is the manifold for the back view as this is the least self occluding view (left most manifold in Fig. 3.2). In this case the manifold can be embedded in a two dimensional space. For other views the curve starts to twist to be a three dimensional space curve. This is primarily because of the similarity imposed by the view point which attracts far away points on the manifold closer. The ultimate twist happens in the side view manifold where the curve twists to be a figure eight shape where each cycle of the eight (half eight) lies in a different plane. Each cycle of the eight figure corresponds to half a walking cycle. The cross point represents the body pose where it is totally ambiguous from the side view to determine from the shape of the contour which leg is in front as can be noticed in Fig. 3.1. Therefore, in a side view, three-dimensional embedding space is the least we can use to discriminate different poses. Embedding a side view cycle in a two-dimensional embedding space results in an embedding similar to that shown in top left of Fig. 3.1 where the two half cycles lies over each other.

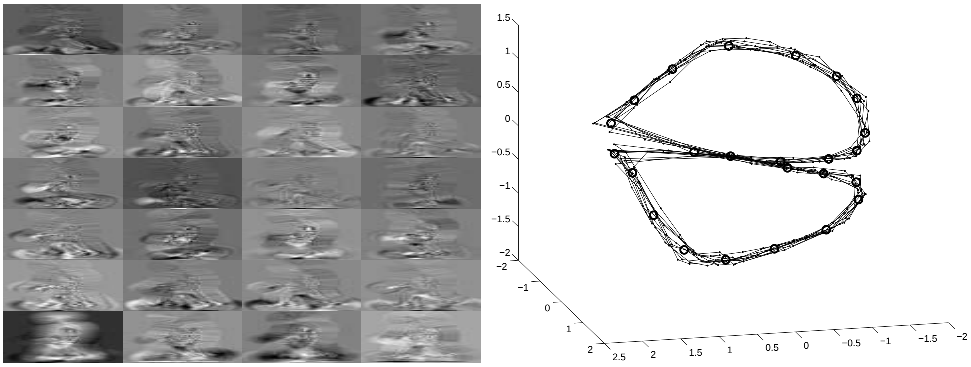


Figure 3.3: Left: Learned mapping coefficients for each of 24 cluster centers along the manifold. The last row represent the learned polynomial coefficients. Right: embedded manifold and cluster centers.

3.4.2 Learning

Given the embedded representation of the manifold in a 3-dimensional Euclidean space, K-means clustering is used to obtain representative points along the embedded manifold. The representative points were used to learn nonlinear mapping from the embedding space to the input space in the form of Eq. 3.5 using TPS kernels. Since the cluster centers do not necessarily coincide with actual data points, generalized RBF interpolation were used, i.e., in the form of Eq. 3.8. Fig. 3.3 shows the learned mapping coefficients and the cluster centers.

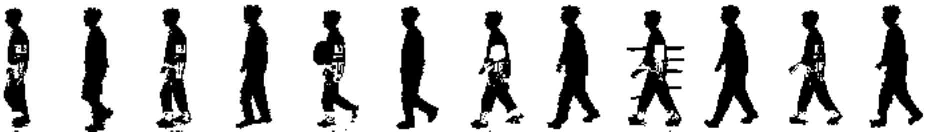


Figure 3.4: Example pose-preserving reconstruction results. Six noisy and corrupted silhouettes and their reconstructions next to them.

Synthesis, Recovery and Reconstruction

Fig. 3.4 shows examples of the reconstruction given corrupted silhouettes as input. In this example, the manifold representation and the mapping were learned from one person data and tested on other people date. Given a corrupted input, after solving for the global geometric transformation, the input is projected to the embedding space using the closed-form inverse mapping approximation in Sec. 3.3.1. The nearest embedded manifold point represents the intrinsic body configuration. A reconstruction of the input can achieved by projecting back to the input space using the direct mapping in Eq. 3.5. As can be noticed from the figure, the reconstructed silhouettes preserve the correct body pose in each case which shows that solving for the inverse mapping yields correct points on the manifold. Notice that no mapping is learned from the input space to the embedded space.

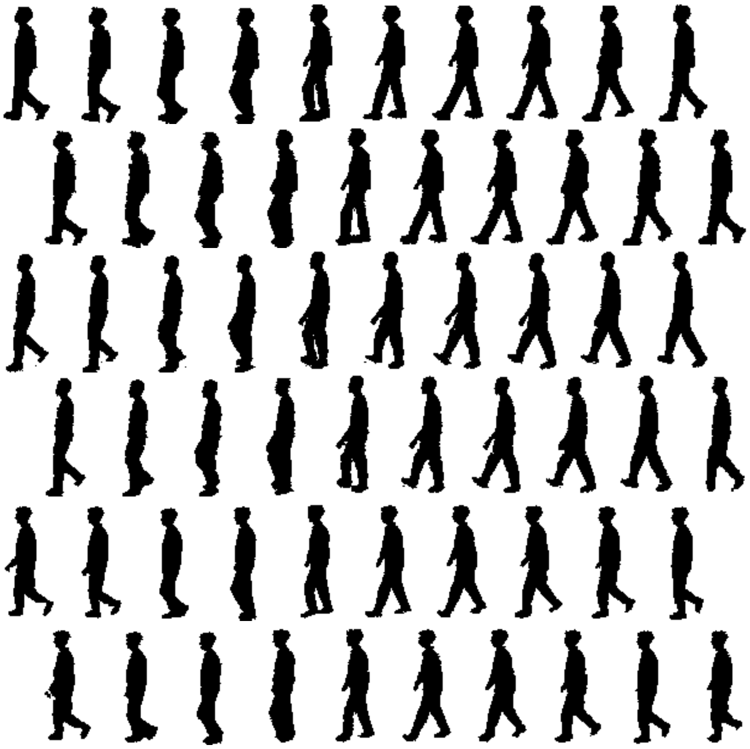
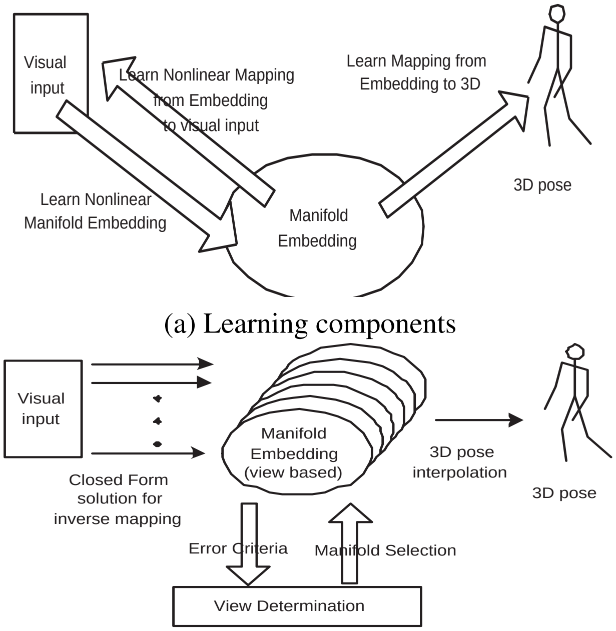


Figure 3.5: Shape synthesis for three different people. First, third and fifth rows: samples used in learning. Second, fourth, sixth rows: interpolated shapes at intermediate configurations (never seen in the learning)

Fig. 3.5 shows an example of shape synthesis and interpolation. Given a learned generative model in the form of Eq. 3.5, we can synthesize new shapes through the walking cycle. In these examples only 10 samples were used to embed the manifold for half a cycle on a unit circle in 2D and to learn the model. Silhouettes at intermediate body configurations were synthesized (at the middle point between each two centers) using the learned model. The learned model can successfully interpolate shapes at intermediate configurations (never seen in the learning) using only two-dimensional embedding. The figure shows results for three different peoples.

3.5 Inferring 3D Body Pose from Silhouettes

In this section we show a direct application of the proposed framework for learning nonlinear motion manifolds in the recovery of 3D body pose. Recovery of 3D body pose is a fundamental problem for human motion analysis in many applications such as motion capture, vision interface, visual surveillance, and gesture recognition. Human body is an articulated object that moves through the three-dimensional world. This motion is constrained by 3D body kinematics and dynamics as well as the dynamics of the activity being performed. Such constraints are explicitly exploited to recover the body configuration and motion in model-based approaches. Alternatively, 3D body pose can be directly inferred from the visual input [55, 18, 110, 109, 94, 45, 116]. We call such approaches learning-based since their objective is to directly infer the 3D body pose as a function of the visual input. Such approaches have great potentials in solving the fundamental initialization problem for model-based vision. The approach we present in this section is inline with the learning-based approaches for pose recovery.


Visual input
Learn Nonlinear Mapping from Embedding to visual input
Learn Mapping from Embedding to 3D
Learn Nonlinear Manifold Embedding
Manifold Embedding
3D pose
Visual input
Manifold Embedding (view based)
Closed Form solution for inverse mapping
3D pose interpolation
3D pose
Error Criteria
Manifold Selection
View Determination

(a) Learning components

(b) pose estimation.

Figure 3.6: Block diagram for the framework. a: Leaning components. b: 3D pose estimation.

Given a visual input (silhouette), the objective is to recover the intrinsic body configuration, recover the view point, reconstruct the input and detect any spatial or temporal outliers. In other words, we aim to simultaneously solve for the pose, view point, and reconstruct the input.

The framework is based on learning three components as shown in Fig. 3.6-a:

1. Learning Manifold Representation: using nonlinear dimensionality reduction we achieve an embedding of the global deformation manifold that preserves the geometric structure of the manifold as described in Sec. 3.2. Given such embedding, the following two nonlinear mappings are learned.

2. Manifold-to-input mapping: a nonlinear mapping from the embedding space into visual input space as described in Sec. 3.3.

3. Manifold-to-pose: a nonlinear mapping from the embedding space into the 3D body pose space.

3.5.1 Determining Intrinsic Configuration

Given a visual input $\boldsymbol{y} \in \mathbb{R}^d$ (silhouette) and given learned manifold representation and manifold-to-input mapping, we can obtain the embedding coordinate $\boldsymbol{x} \in \mathbb{R}^e$ corresponding to the input in a closed-form as was shown in Sec. 3.3.1. The recovered point $\boldsymbol{x}$ is typically enough to recover the pose. However to enhance the result and constrain the solution, we need to find the closest manifold point, which can also be obtained efficiently. For the gait case, the manifold is one dimensional, and therefore, only one dimensional search is sufficient to recover the manifold point closest to the input. To obtain such point, the embedded manifold is fitted with a cubic spline $m(t)$ as a function of the time variable $t \in [0, 1]$ where each cycle of the activity is temporally mapped from 0 to 1. Given such model, a one dimensional search is used to obtain t^* that minimizes $\|\boldsymbol{x} - m(t)\|$. Reconstruction can be achieved by re-mapping the projected point using Eq. 3.5.

3.5.2 Determining View Point

Given the learned view-based manifolds $\boldsymbol{M}_v$ and the learned view-based mappings $\boldsymbol{B}_v \psi_v(\boldsymbol{x})$ for each view v, determining the view point reduces to finding the manifold that minimizes the inverse-mapping error of an input $\boldsymbol{y}$ or a sequence of inputs $\boldsymbol{y}_t$. Given an input $\boldsymbol{y}$ and its projections $\boldsymbol{x}_v^*$ into each manifold we chose the manifold that minimizes

$$\|\boldsymbol{x}_v^* - m_v(t_v*)\|,$$

where t_v is the corresponding spline parameter. Fig. 3.7 shows five view manifolds and the projection of a sequence to the five manifolds.

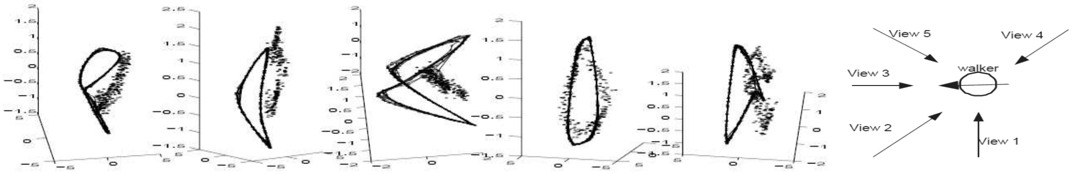


Figure 3.7: Five manifolds for five view points and the projection of a sequences to each manifold.

3.5.3 Learning Mapping: Manifold-to-3D

Similar to the mapping from the embedding space into the visual input, a mapping can be learned from the embedding space to the 3D body joint space. RBF interpolants in the form of Eq. 3.4 between the embedding space R^e and each degree of freedom of each body joint. We represent the body using 16 joints model and each joint is represented by its coordinates in a body centered global coordinate system. Representative points on the manifolds as well as their corresponding 3D body configurations are used in order to learn the mapping parameters as was shown in Sec. 3.3.

3.6 Experimental Results

3.6.1 3D Pose Recovery Results

Validation Experiment

In order to validate that our approach can interpolate 3D poses from input silhouettes, we used a sequence from Georgia tech gait data with ground truth provided by motion capture data. the sequence contains 72 frames where we learn the model using the odd numbered frames and evaluated on the even numbered frames. The resulted 3D reconstruction is compared to the ground truth and is plotted in Fig. 3.8 for four of the sixteen joint angles (right foot, left foot, lower right leg, lower left leg).

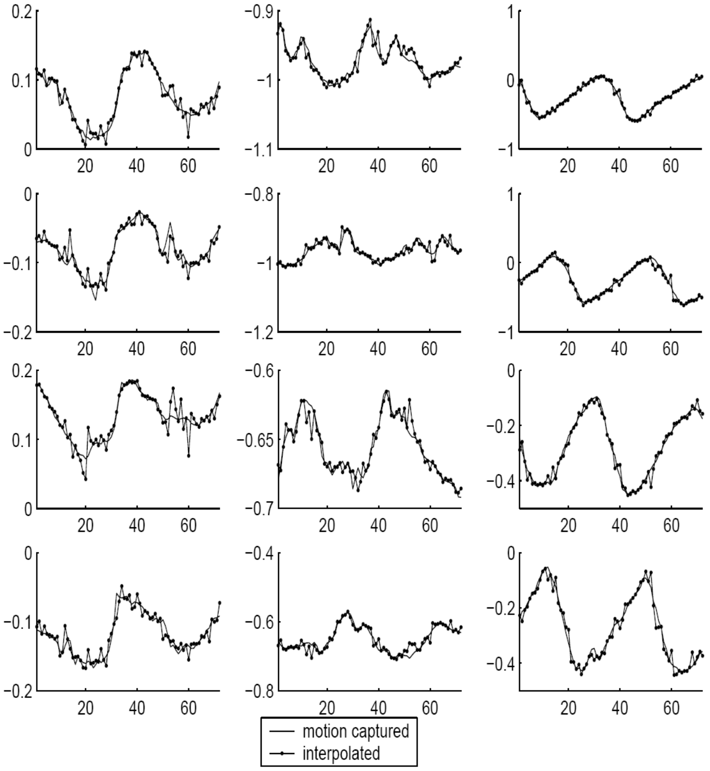


Figure 3.8: Evaluation of 3D reconstruction with ground truth for four joints (right foot, left foot, Lower right leg, lower left leg). Each row represents a joint angle x,y,z. (units in foot)

Generalization

In order to show that the approach generalizes to different people, we used the CMU MoboGait database to train and evaluate the proposed approach. Each sequence of the database contains about 300 frames (8-11 walking cycles). The database contains 6 views of each walking person. We used five of them. The used views are shown in Fig. 3.7.

In each experiment, we used one person sequences to learn the manifolds of the five views and the mappings from the manifolds to the input sequences. The mappings from each of the manifolds to 3D body configuration were also learned. For the evaluation we use other people's sequences to evaluate the 3D reconstruction [2]. Fig. 3.9 shows the view classification

[2] For the experiment we show here we use person 37 for the learning and evaluate on persons 15 in Fig. 3.10 and on 70, 86, 76, 79 in Fig. 3.11

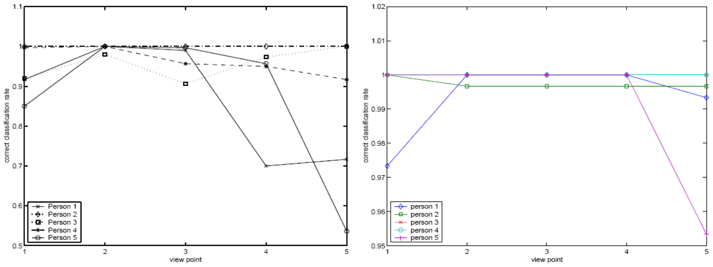


Figure 3.9: View classification: a- classification from single frames. b- classification with boosting multiple frames

results for five evaluation sequences (five people) and five views. Overall correct classification rate is 93.05%. Obviously the view classification from a single frame can be erroneous because of self occlusion and therefore boosting several frames would lead to better results which is shown in Fig. 3.9-b where majority vote were used over view classification results for each sequence of five frames which results in a correct classification rate of 99.63%.

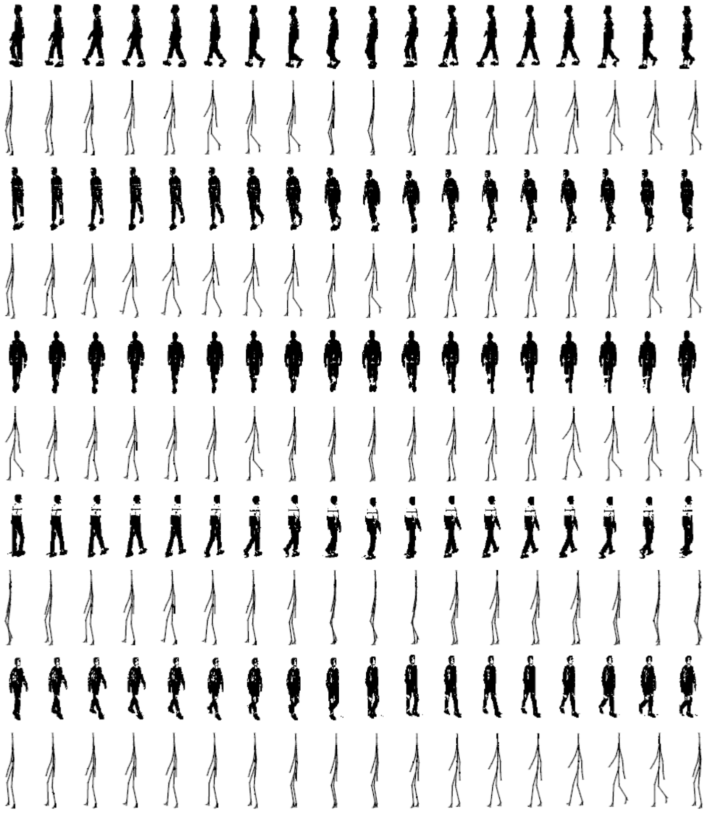


Figure 3.10: 3D reconstruction for five views.

Fig. 3.10 shows the 3D reconstruction for one person for each of the five views. Since the input sequences are synchronized, the reconstructed 3D poses from each view are supposed to be the same. The 3D reconstructions are always shown from the side view point. The reconstruction shows qualitatively correct reconstruction from all views. Unfortunately, there are no ground truth to evaluate the results of this experiment. Fig. 3.11 shows some 3D reconstruction results for four other people. As can be noticed, the input silhouettes are noisy.

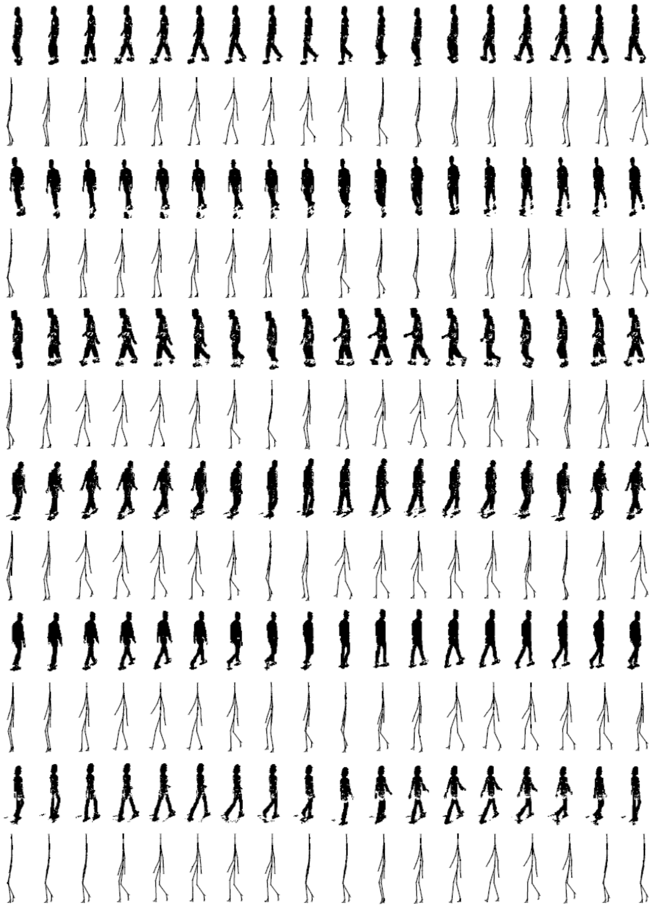


Figure 3.11: 3D reconstruction for 4 people from different views: From top to bottom: person 70 views 1,2; person 86 views 1,2; person 76 view 4; person 79 view 4

Fig. 3.12 shows 3D pose reconstructed from corrupted silhouette which are typical in surveillance applications due to errors in background subtraction, shadows, fragmentation, and carried objects. Reconstruction of the input silhouettes can be achieved by mapping back to the input space.

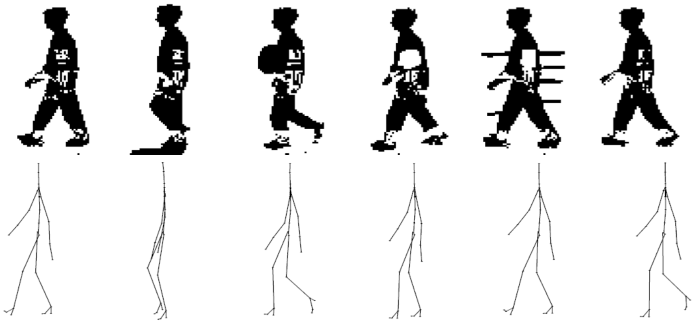


Figure 3.12: 3D reconstruction from corrupted inputs

3.6.2 Dynamic Appearance Examples

In this section we show two examples for learning manifolds of dynamic appearance. In the first example, we learn the model for two arm gestures (raising arm up and down in two different way) as shown in Fig. 3.13 . Four cycles for each of the two gestures (total of 402 frames) were used to embed the manifold and learn a model in the form of Eq. 3.5 using centers set at approximately equal distance along the embedded manifolds. As expected, each of these gestures is 1-dimensional manifold and a 2-dimensional embedding space is enough to discriminate them. Fig. 3.13 show the embedding and the centers. Given the learned model, recovery of the embedding coordinate was achieved using the closed-form inverse mapping approximation as was shown in Sec. 3.3.1. Fig. 3.13 also shows the recovered embedding coordinates.

Fig. 3.14 shows the results for learning the manifold for face motion during a smile. The sequence contains 44 frames from the CMU facial expression dataset. Obviously this is a one dimensional manifold. Embedding in this case was done in a one dimensional space, i.e., samples were embedded on a line. The model was learned using approximately equally-spaced center on the manifold. The embedding and the centers is shown in Fig. 3.14-c. The learned model was used to synthesize faces at intermediates points along the manifold. The results are shown in Fig. 3.14-a using both Gaussian kernels and TPS kernels. Notice that this result is obtained with only one-dimensional embedding of the manifold and the face is parameterized with only one parameter and still we can reconstruct the original faces.

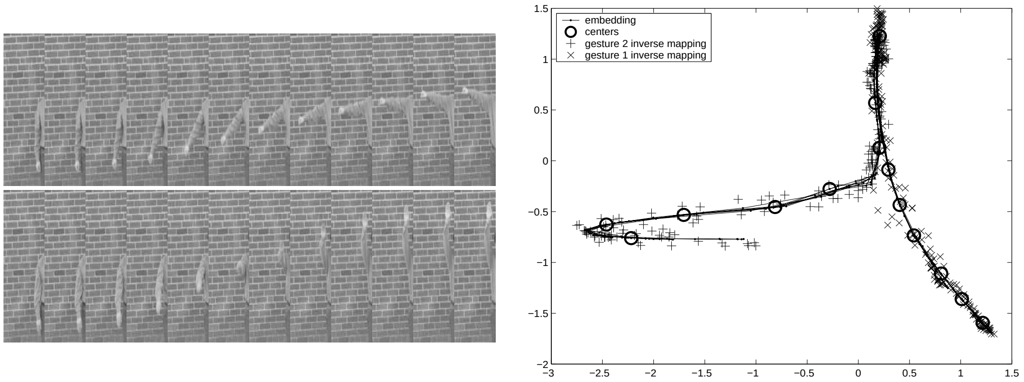


Figure 3.13: Learning two arm gestures. Top: centers. Bottom: Embedding in 2D and inverse mapping results

3.7 Summary

In this chapter we introduced a framework for learning a landmark-free correspondence-free global representations of dynamic shape and dynamic appearance manifolds. The framework is based on using nonlinear dimensionality reduction to achieve an embedding of the global deformation manifold which preserves the geometric structure of the manifold. Given such embedding, a nonlinear mapping is learned from such embedded space into visual input space using RBF interpolation. Given this framework, any visual input is represented by a linear combination of nonlinear bases functions centered along the manifold in the embedded space. In a sense, the approach utilizes the implicit correspondences imposed by the global vector representation which are only valid locally on the manifold through explicit modeling of the manifold and RBF interpolation where closer points on the manifold will have higher contributions than far away points. We also showed how approximate solution for the inverse mapping can be obtained in a closed form which facilitates recovery of the intrinsic body configuration. The framework was applied to learn a representation of the gait manifold as an example of a dynamic shape manifold. We showed how the learned representation can be used to interpolate intermediate body poses as well as in recovery and reconstruction of the input. We extended the approach to learn mappings from the embedded motion manifold to 3D joint angle representation which yields an approximate closed-form solution for 3D pose recovery. We also showed examples of using the framework in learning manifolds for some simple gestures and facial expressions as examples of dynamic appearance manifolds. In Chapter 4, we extended

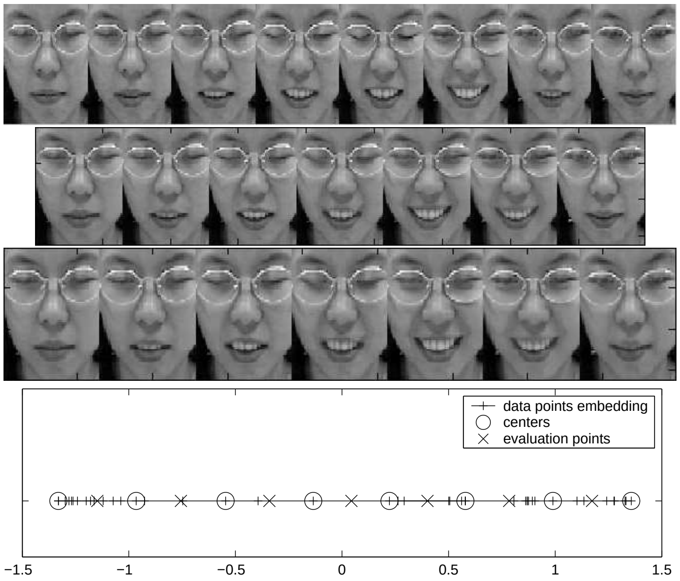


Figure 3.14: Learning a facial expression. Left-top, centers equally spaced on the manifold. Center & bottom: interpolated faces at intermediate points. Right-top, Learned mapping coefficients for the eight centers (the last two are the linear polynomial coefficients). Right-bottom: One dimensional embedding.

the framework to learn a decomposable generative model that separates appearance variations from the intrinsics underlying dynamics manifold though introducing a framework for separation of style and content on a nonlinear manifold.

Chapter 4

Generalized Separation of Style and Content on Motion Manifolds

The problem of separation of style and content is essential task in visual perception and is a fundamental mystery of perception. Such problem appears extensively in different computer vision applications. The problem we address in this chapter is the separation of style and content when the content lies on low dimensional nonlinear manifold representing dynamic object. We show that such setting appears in many human motion analysis problems and therefore we introduce a framework for learning parameterization of style and content in such settings. The framework we present in this paper is based on decomposing the style parameters in the space of nonlinear functions which map between a learned unified embedding of multiple content manifolds and the visual input space. 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.

4.1 Factorized Generative Models

Our objectives is to learn representations for the shape and/or the appearance of moving (dynamic) objects that supports tasks such as synthesis, pose recovery, view recovery, input reconstruction, and tracking. Such learned representation will serve as decomposable generative models for dynamic appearance where we can think of the image appearance (similar argument for shape) of a dynamic object as instances driven from such generative model. 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 view point, illumination, etc. In this paper, we refer to the intrinsic body configuration as the content and all other factors as style factors.

4.1.1 Style and Content Decomposition

We start with the case of factorizing one style factor. Given a set of image sequences, similar to the ones in Fig. 1.2, representing certain motion such as gesture, facial expression, or activity, where each sequence is performed by one subject, we aim to learn a generative model that explicitly decomposes the following two factors:

1. Content (body pose): A representation of the intrinsic body configuration through the motion as a function of time that is invariant to the person, i.e., the content characterizes the motion or the activity.

2. Style (people) : Time-invariant person parameters that characterize the person appearance or shape.

Fig. 1.2 shows an example of such data where different people are performing the same activity as gait or smile motion. The content in this case is the gait motion or the smile motion while the style is the person shape or face appearance, respectively. On the other hand, given an observation of a certain person at a certain body pose and given the learned generative model, we aim to be able to solve for both the body configuration representation (content) and the person parameter (style).

We learn a view-based generative model in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t^c; \boldsymbol{a}, \boldsymbol{b}^s) \ , \tag{4.1}$$

where the image, $\boldsymbol{y}_t$, at time t is an instance driven from a generative model where the function $\gamma(\cdot)$ is a mapping function that maps from a representation of body configuration $\boldsymbol{x}_t^c$ (content) at time t into the image space given mapping parameters $\boldsymbol{a}$ and style dependent parameter $\boldsymbol{b}^s$ that is time invariant. In our case the content is a continuous domain while style is represented by the discrete style classes which exist in the training data where we can interpolate intermediate styles and/or intermediate contents.

Suppose that we can learn a unified, style-invariant, nonlinearly embedded representation of the motion manifold $\mathcal{M}$ in a low dimensional Euclidean embedding space, $\mathbb{R}^e$, then we can learn a set of style-dependent nonlinear mapping functions from the embedding space into the input space, i.e., functions $\gamma_s(\boldsymbol{x}_t^c) \ : \ \mathbb{R}^e \ \rightarrow \ \mathbb{R}^d$ that maps from embedding space with dimensionality e into the input space (observation) with dimensionality d for style class s. Since we consider nonlinear manifolds and the embedding is nonlinear, the use of nonlinear mapping is necessary. In this paper we consider mapping functions of the form

$$\boldsymbol{y}_t = \gamma_s(\boldsymbol{x}_t^c) = \boldsymbol{C}^s \cdot \psi(\boldsymbol{x}_t^c) \ , \tag{4.2}$$

where $\boldsymbol{C}^s$ is a $d \times N$ linear mapping and $\psi(\cdot) : \mathbb{R}^e \rightarrow \mathbb{R}^N$ is a nonlinear mapping where N basis functions are used to model the manifold in the embedding space, i.e.,

$$\psi(\cdot) = [\phi_1(\cdot), \cdots, \phi_N(\cdot)]^T \tag{4.3}$$

Given learned models of the form of Eq. 4.2, the style can be decomposed in the linear mapping coefficient space using bilinear model in a way similar to [128]. Therefore, an input instance $\boldsymbol{y}_t$ can be written as asymmetric bilinear model in the linear mapping space as

$$\boldsymbol{y}_t = \boldsymbol{\mathcal{A}} \times_3 \boldsymbol{b}^s \times_2 \psi(\boldsymbol{x}_t^c) \tag{4.4}$$

where $\boldsymbol{\mathcal{A}}$ is a third order tensor (3-way array) with dimensionality $d \times N \times J$, $\boldsymbol{b}^s$ is a style vector with dimensionality J, and $\times_n$ denotes mode-n tensor product.

A challenging task to achieve such decomposition is to learn a unified and style-invariant embedded representation of the motion manifold. Several approaches can be used to achieve such representation. We used LLE to obtain manifold embedding for each individual sequence as described in Sec. 3.4.1. A mean manifold is computed as a unified representation through nonlinear warping of manifold points in Sec. 4.2.

4.1.2 Multiple Style Factor Decomposition

We extend the style and content factorization to 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 model in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t; \boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n) \tag{4.5}$$

where the appearance, $\boldsymbol{y}_t$, at time t is an instance driven from a generative model where the function $\gamma(\cdot)$ is a mapping function that maps body configuration $\boldsymbol{x}_t$ at time t into the image

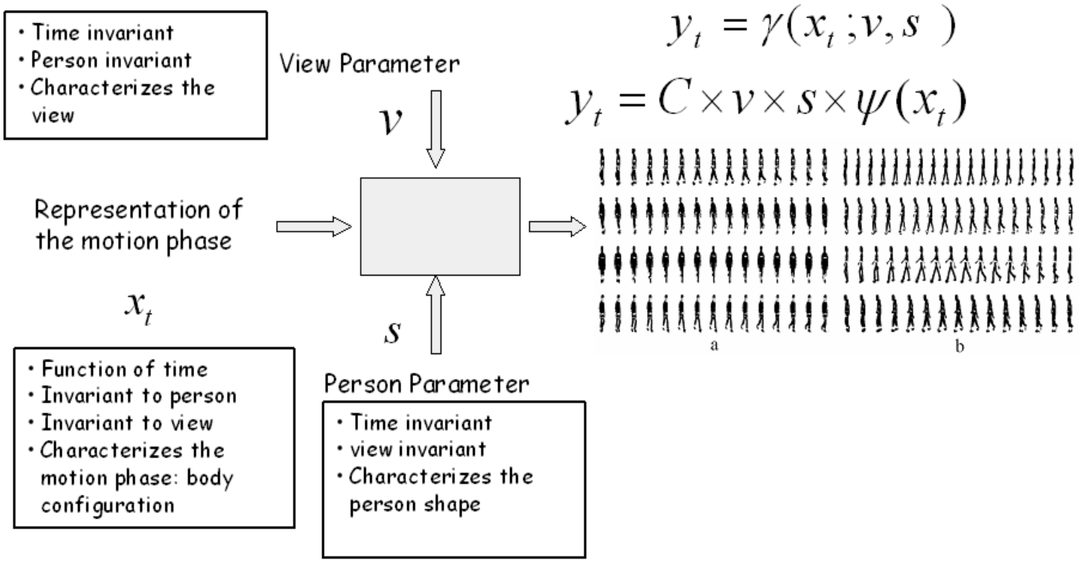

• Time invariant
• Person invariant
• Characterizes the view
View Parameter
v
Representation of the motion phase
x_t
• Function of time
• Invariant to person
• Invariant to view
• Characterizes the motion phase: body configuration
s
Person Parameter
• Time invariant
• view invariant
• Characterizes the person shape
$y_t = \gamma(x_t; v, s)$
$y_t = C \times v \times s \times \psi(x_t)$
a
b

Figure 4.1: 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.

space. i.e., the mapping function γ maps from a representation of the body configuration space into the image space given mapping parameters $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n$ each representing a set of conceptually orthogonal factors. Such factors are independent of the body configuration and can be time variant or invariant. The general form for the mapping function $\gamma(\cdot)$ that we use is

$$\gamma(\boldsymbol{x}_t; \boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n) = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{b}_1 \times \cdots \times_n \boldsymbol{b}_n \cdot \psi(\boldsymbol{x}_t) \tag{4.6}$$

where $\psi(\boldsymbol{x})$ is a nonlinear kernel map from a representation of the body configuration to a kernel induced space and each $\boldsymbol{b}_i$ is a vector representing a parameterization of orthogonal factor i, $\boldsymbol{\mathcal{A}}$ is a core tensor, $\times_i$ is *mode-i* tensor product as defined in [72]. In the model in Eq. 4.6, the relation between body configuration and the input is nonlinear where other factors are approximated linearly through high-order tensor analysis. The model in Eq. 4.6 is a generalization over the style and content model in Eq. 4.4, where only one factor can be decomposed. In the model in Eq. 4.6, the relation between body configuration and the input is nonlinear where other factors are approximated linearly through multilinear high-order tensor analysis.

For example, for the gait case (as shown in Fig. 4.1), if we consider multiple views in addition to multiple people, the data set have three components: style, view, in addition to body configuration. A generative model for walking silhouettes for different people from different view points will be in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t; \boldsymbol{v}, \boldsymbol{s}) = \boldsymbol{\mathcal{A}} \times \boldsymbol{v} \times \boldsymbol{s} \times \psi(\boldsymbol{x}_t) \tag{4.7}$$

where $\boldsymbol{v}$ is a parameterization of the view, which is independent of the body configuration but can change over time, and $\boldsymbol{s}$ is a parameterization of the shape style of the person performing the walk which is independent of the body configuration and time invariant. The body configuration $\boldsymbol{x}_t$ evolves along a representation of the manifold that is homeomorphic to the actual gait manifold.

In the following sections we describe the details for fitting such models and estimation of the parameters. Section 4.2 describes how to obtain a unified nonlinear embedding of the motion manifold for style analysis. Section 4.3 describes model learning and solving for style and content factors. Sections 4.4 and 4.5 describe the generalized model and solving for multiple factors.

4.2 Content Manifold Embedding:Embedding Multiple Manifolds

4.2.1 Embedding Multiple Manifolds

Given sequences for multiple people, we need to obtain a unified embedding for the underlying body configuration manifold. Nonlinear dimensionality reduction approaches such as [126, 111, 19] are not able to embed multiple people manifolds simultaneously. Although such approaches try to capture the manifold geometry, typically, the distances between instances of the same person (within the same manifold) is much smaller than distances between corresponding points on different people's manifolds. Therefore, they tend to capture the intrinsic structure of each manifold separately without generalizing to capture inter-manifolds aspects. This is shown in Fig. 4.2 (a) where LLE is used to embed three people's manifolds 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-distance.). Even if we force LLE to include corresponding points on different manifolds to each point's neighbors, this results in artificial embedding that does not capture the manifold geometry. Another fundamental problem is that different people will have different manifolds because the appearance (shape) is different, which imposes different twists to the manifolds and therefore different geometry. This can be noticed in Fig. 4.2 (b).

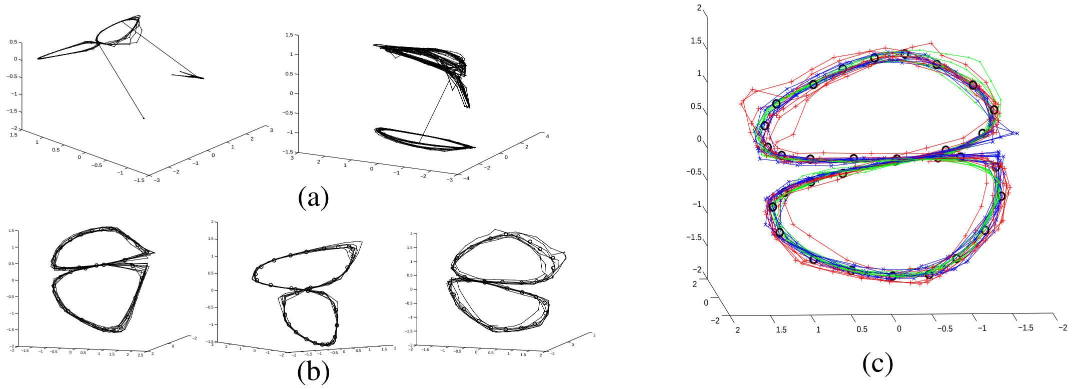

(a)
(b)
(c)

Figure 4.2: Multiple manifold embedding: (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 $\tilde{X}^k$

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 points are temporal-mapped from 0 to 1. For the case of periodic motion, such as gait, each cycle on the manifold is time warped 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 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., $\boldsymbol{m}^k(t) : t \rightarrow \mathbb{R}^e$ where $t = 0 \rightarrow 1$ is the time variable. The manifold for person k is sampled at N uniform time instances $\boldsymbol{m}^k(t_i)$ where $i = 1 \cdots N$.

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 [25]. We solve for a mean manifold $Z(t_i)$ and a set of non-rigid transformations $f(.; \alpha_k)$ where the objective is to minimize the energy function

$$E(f) = \sum_k \sum_i ||Z(t_i) - f(\boldsymbol{m}^k(t_i); \alpha_k)||^2 + \lambda ||Lf||^2 \qquad (4.8)$$

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

$$\tilde{\boldsymbol{X}}^k = f(\boldsymbol{X}^k; \alpha_k), k = 1 \cdots K. \tag{4.9}$$

Fig. 4.2 (b),(c) shows an example of three different manifolds and their warping into a unified manifold embedding. When there are multiple variant factors, however, the data driven each individual manifold is quite different and hard to find unified representation using non-rigid transformation.

4.3 Decomposition

4.3.1 Learning Style Dependent Mappings

Let the sets of input image sequences be $\mathsf{Y}^k = \{\boldsymbol{y}_i^k \in \mathbb{R}^d \quad i = 1, \cdots, N_k\}$ and let their corresponding points on the unified embedding space be $\mathsf{X}^k = \{\boldsymbol{x}_i^k \in \mathbb{R}^e, \quad i = 1, \cdots, N_k\}$ where e is the dimensionality of the embedding space (e.g. $e = 3$ in the case of gait) and $k = 1 \cdots K$ is the person (style) index. Let the set of N centers representing the mean manifold be $Z = \{\boldsymbol{z}_j \in \mathbb{R}^e, j = 1, \cdots, N\}$. We can learn nonlinear mappings between the centers Z and each of the input sequence using generalized radial basis function interpolation (GRBF) [103], i.e., one mapping for each style class k. We can solve for multiple interpolants $f^l : \mathbb{R}^e \rightarrow \mathbb{R}$ as described in Sec. 3.3.

$$f^l(\boldsymbol{x}) = p^l(\boldsymbol{x}) + \sum_{i=1}^{N} w_j^l \phi(|\boldsymbol{x} - \boldsymbol{z}_j|), \tag{4.10}$$

where $\phi(\cdot)$ is a real-valued basic function, w_j are real coefficients, $|\cdot|$ is the norm on $\mathbb{R}^e$ (the embedding space). The whole mapping can be written in a matrix form as

$$f_k(\boldsymbol{x}) = \boldsymbol{C}^k \cdot \psi(\boldsymbol{x}), \tag{4.11}$$

where $\boldsymbol{C}^k$ is a $d \times (N+e+1)$ dimensional matrix with the l-th row $[w_1^l \cdots w_N^l \quad c^{l^\top}]$ and the vector $\psi(\boldsymbol{x})$ is $[\phi(|\boldsymbol{x} - \boldsymbol{z}_1|) \cdots \phi(|\boldsymbol{x} - \boldsymbol{z}_N|) \;\; 1 \;\; \boldsymbol{x}^\top]^\top$. The matrix $\boldsymbol{C}^k$ represents the coefficients for d different nonlinear mappings for style class k.The solution for $\boldsymbol{C}^k$ can be obtained by directly solving the linear systems as explained in Sec. 3.3.

4.3.2 Separating Style

Given learned nonlinear mapping coefficients $\boldsymbol{C}^1, \boldsymbol{C}^2, \cdots, \boldsymbol{C}^K$ for each person, the style parameters can be decomposed by fitting an asymmetric bilinear model [128] to the coefficient tensor. Let the coefficients be arranged as a $d \times M \times K$ tensor $\boldsymbol{\mathcal{C}}$, where $M = (N + e + 1)$. Therefore, we are looking for a decomposition in the form

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}}^c \times_3 \boldsymbol{B}^s$$

where $\boldsymbol{\mathcal{A}}^c$ is $d \times M \times J$ tensor containing content bases for the RBF coefficient space and $\boldsymbol{B}^s = [\boldsymbol{b}^1 \cdots \boldsymbol{b}^K]$ is a $J \times K$ style coefficients. This decomposition can be achieved by arranging the mapping coefficients as a $dM \times K$ matrix as

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{c}_1^1 & \cdots & \boldsymbol{c}_1^K \\ \vdots & \ddots & \vdots \\ \boldsymbol{c}_M^1 & \cdots & \boldsymbol{c}_M^K \end{pmatrix} \tag{4.12}$$

where $[\boldsymbol{c}_1^k, \cdots, \boldsymbol{c}_M^k]$ are the columns for RBF coefficients $\boldsymbol{C}^k$. Given the matrix $\boldsymbol{C}$ style vectors and contents bases can be obtained by singular value decomposition as $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ where the content bases are the columns of $\boldsymbol{U}\boldsymbol{S}$ and the style vectors are the rows of $\boldsymbol{V}$.

4.3.3 Solving for Style and Content

Given a 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 orthogonal factors as well as body configuration. We first present EM-like iterative solution for estimation of style and content factors.

Given a new input $\boldsymbol{y} \in \mathbb{R}^d$, it is required to find both the content, i.e., the corresponding embedding coordinates $\boldsymbol{x} \in \mathbb{R}^e$ on the manifold, and the person style parameters $\boldsymbol{b}^s$. These parameters should minimize the reconstruction error defined as

$$E(\boldsymbol{x}^c, \boldsymbol{b}^s) = ||\boldsymbol{y} - \boldsymbol{\mathcal{A}} \times \boldsymbol{b}^s \times \psi(\boldsymbol{x}^c)||^2 \tag{4.13}$$

Solving for content

If the style vector, $\boldsymbol{b}^s$, is known, we can solve for the content $\boldsymbol{x}^c$. Note that, in our case, the content is a continuous variable in a nonlinearly embedded space. A solution for $\boldsymbol{x}^*$ can be obtained by least square solution for the over-constrained nonlinear system $\boldsymbol{x}^* = \arg_{\boldsymbol{x}} \min ||\boldsymbol{y} - \boldsymbol{B}\psi(\boldsymbol{x})||^2$ where $\boldsymbol{B} = \boldsymbol{\mathcal{A}} \times \boldsymbol{b}^s$. We show here how to obtain a closed-form solution for $\boldsymbol{x}^c$ in Sec. 3.3.1.

Solving for style

If the embedding coordinate (content) is known, we can solve for style vector $\boldsymbol{b}_s$. Given style classes $\boldsymbol{b}^k$, $k = 1, \cdots, K$ learned from the training data and given the embedding coordinate $\boldsymbol{x}$, the observation can be considered as drawn from a Gaussian mixture model centered at $\boldsymbol{\mathcal{A}} \times \boldsymbol{b}^k \times \psi(\boldsymbol{x})$ for each style class k. Therefore, observation probability $p(\boldsymbol{y}|k, \boldsymbol{x})$ can be computed as

$$p(\boldsymbol{y}|k, \boldsymbol{x}) \propto \exp -||\boldsymbol{y} - \boldsymbol{\mathcal{A}} \times \boldsymbol{b}^k \times \psi(\boldsymbol{x})||^2/(2\sigma^2). \tag{4.14}$$

Style conditional class probabilities can be obtained as

$$p(k|\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y}|k, \boldsymbol{x})p(k|\boldsymbol{x})/p(\boldsymbol{y}|\boldsymbol{x}) \tag{4.15}$$

where $p(\boldsymbol{y}|\boldsymbol{x}) = \sum_k p(\boldsymbol{y}|\boldsymbol{x}, k)p(k)$. A new style vector can then be obtained as a linear combination of the K class style vectors as

$$\boldsymbol{b}^s = \sum_k w_k \boldsymbol{b}^k \tag{4.16}$$

where the weights w_k are set to be $p(k|\boldsymbol{x}, \boldsymbol{y})$. Given the two steps described above we can solve for both style $\boldsymbol{b}^s$ and content $\boldsymbol{x}^c$ in an EM-like iterative procedure where in the E-step we calculate the content $\boldsymbol{x}^c$ given the style parameters and in the M-step we calculate new style parameters $\boldsymbol{b}^s$ given the content. The initial content can be obtained using a mean style vector $\tilde{\boldsymbol{b}}^s$.

4.4 Generalized Style factorization

In Sec. 4.3 it was shown how to separate a style factor when learning a generative model for data lying on a manifold. Here we generalize this concept to factorize several style factors.

For example, consider the walking motion observed from multiple view points (as silhouettes). The resulting data lie on multiple subspaces and/or multiple manifolds. There is the underling motion manifold, which is one dimensional for the gait motion. Besides the motion, there is the view manifold and the space of different people's shapes. Another example we consider is facial expressions. Consider face data of different people performing different facial dynamic expressions such as sad, smile, surprise, etc. The resulting face data posses several dimensionality of variability: the dynamic motion, the expression type and the person face. So, how to model such data in a generative manner. We follow the same framework of explicitly modeling the underlying motion manifold and over that we decompose various style factors.

We can think of the image appearance (similar argument for shape) of a dynamic object as instances driven from such generative model. 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 model in the form

$$\boldsymbol{y}_t = \mathcal{A} \times_1 \boldsymbol{b}_1 \times \cdots \times_n \boldsymbol{b}_n \cdot \psi(\boldsymbol{x}_t) \tag{4.17}$$

where $\psi(\boldsymbol{x})$ is a nonlinear kernel map from a representation of the body configuration to a kernel induced space and each $\boldsymbol{b}_i$ is a vector representing a parameterization of orthogonal factor i, $\mathcal{A}$ is a core tensor, $\times_i$ is *mode-i* tensor product as defined in [72].

For example for the gait case, a generative model for a walking silhouettes for different people from different view points will be in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t; \boldsymbol{v}, \boldsymbol{s}) = \mathcal{A} \times \boldsymbol{v} \times \boldsymbol{s} \times \psi(\boldsymbol{x}_t) \tag{4.18}$$

where $\boldsymbol{v}$ is a parameterization of the view, which is independent of the body configuration but can change over time, and $\boldsymbol{s}$ is a parameterization of the shape style of the person performing the walk which is independent of the body configuration and time invariant. The body configuration $\boldsymbol{x}_t$ evolves along a representation of the gait manifold. The question is how to obtain such representation of the gait manifold that is invariant to different people shape styles and different views.

Another example is modeling the manifolds of facial expression motions. Given a dynamic facial expression such as sad, surprise, happy, etc., where each expression start from neutral and evolve 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 be in the form

$$\boldsymbol{y}_t = \gamma(\boldsymbol{x}_t; \boldsymbol{e}, \boldsymbol{f}) = \mathcal{A} \times \boldsymbol{e} \times \boldsymbol{f} \times \psi(\boldsymbol{x}_t) \tag{4.19}$$

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

4.4.1 Homeomorphic Manifold Analysis

The model in Eq. 4.17 is a generalization over the model in Eq. 4.4. However, such generalization is not obvious. In Sec. 4.2, LLE was used to obtain manifold embeddings, and then a mean manifold is computed as a unified representation through nonlinear warping of manifold points. However, since the manifolds twists 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. These limitations motivate the use of a conceptual unified representation of the configuration manifold that is independent of all other factors. Such unified representation would allow the model in Eq. 4.17 to generalize to decompose as many factors as desired. In this model, the relation between body configuration and the input is nonlinear where other factors are approximated linearly through multilinear analysis. The use of nonlinear mapping is essential since the embedding of the configuration manifold is nonlinearly related to the input. Since the model is generative (from embedding to visual input) and nonlinear mapping is used, any representation can be used to model the content manifold as long as it is homeomorphic to the actual manifold.

For example for the gait case in Eq. 4.18, the body configuration $\boldsymbol{x}_t$ can evolve along a conceptual representation of the manifold that is homeomorphic to the actual gait manifold. The question is what conceptual representation of the manifold we can use. Since the gait is one dimensional closed manifold embedded in the input space, it is homeomorphic to a unit circle embedded in 2D. In general, all closed 1D manifolds are topologically homeomorphic to a unit circle. We can think of it as a circle twisted and stretched in the space based on the shape and the appearance of the person under consideration or based on the view. 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 unit circle, the actual data is used to learn nonlinear warping between the conceptual representation and the actual data manifold. Since each manifold will have its own mapping, we need to have a mechanism to parameterize such mappings and decompose all these mappings to parameterize variables for views, different people, etc.

4.4.2 Separating Multiple Factors

Without lose of generality, we will use the gait model in Eq. 4.18 as an example in this section, while fitting more factors are straight forward generalization.

The input is a set of image sequences each represents a full cycle of the motion, e.g., a full walking cycle captured from different view points. Each image sequence is of certain person and certain view. We assume that the view does not change within any sequence. Each person can have multiple image sequences. The image sequences are not necessarily of the same length. We denote each sequence by $\boldsymbol{Y}^{sv} = \{\boldsymbol{y}_1^{sv} \cdots \boldsymbol{y}_{N_{sv}}^{sv}\}$ where v denotes the view class index and s is style index. Let N_v and N_s denote the number of views and number of styles respectively, i.e., there are $N_s \times N_v$ sequences. Each sequence is temporally embedded at equidistance on a unit circle such that $\boldsymbol{x}_i^{sv} = [cos(2\pi i/N_{sv} + \delta^{sv}) \; sin(2\pi i/N_{sv} + \delta^{sv})], i = 1 \cdots N_{sv}$ where the displacement parameter δ is used to align all the embedded sequences. Notice that by temporal embedding on a unit circle we do not preserve the metric in input space. Rather, we preserve the topology of the manifold.

Given a conceptual content manifold embedding obtained, we can learn nonlinear style-dependent mappings for each of the style factors. Given a set of distinctive representative and arbitrary points $\{\boldsymbol{z}_i \in \mathbb{R}^2, i = 1 \cdots N\}$ we can define an empirical kernel map [113] as

$\psi_N(x) : \mathbb{R}^2 \rightarrow \mathbb{R}^N$ where

$$\psi_N(\boldsymbol{x}) = [\phi(\boldsymbol{x}, \boldsymbol{z}_1), \cdots, \phi(\boldsymbol{x}, \boldsymbol{z}_N)]^{\mathsf{T}}, \tag{4.20}$$

given a kernel function $\phi(\cdot)$. For each input sequence $\boldsymbol{Y}^{sv}$ and its embedding $\boldsymbol{X}^{sv}$ we can learn a nonlinear mapping function $f^{sv}(\boldsymbol{x})$ that satisfies $f^{sv}(\boldsymbol{x}_i) = \boldsymbol{y}_i, i = 1 \cdots N_{sv}$ and minimizes a regularized risk criteria. From the representer theorem, such function admits a representation of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \phi(\boldsymbol{x}, \boldsymbol{z}_i),$$

i.e., the whole mapping can be written as

$$f^{sv}(\boldsymbol{x}) = \boldsymbol{C}^{sv} \cdot \psi(\boldsymbol{x}) \tag{4.21}$$

where $\boldsymbol{C}$ is a $d \times N$ coefficient matrix. As described in Sec. 3.3, the mapping coefficients can be obtained by solving the linear system

$$[\boldsymbol{y}_1^{sv} \cdots \boldsymbol{y}_{N_{sv}}^{sv}] = \boldsymbol{C}^{sv}[\psi(\boldsymbol{x}_1^{sv}) \cdots \psi(\boldsymbol{x}_{N_{sv}}^{sv})]$$

.

To align the sequences we use the model learned for a prototype cycle as a reference. Given a prototype cycle coefficients C^*, any new cycle embedding coordinate is aligned to it by searching for the displacement parameter δ that minimizes the reconstruction error

$$E(\delta) = \sum_i \|\boldsymbol{y}_i - \boldsymbol{C}^* \cdot \psi(\boldsymbol{x}_i(\delta))\| \tag{4.22}$$

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 [128]. Higher-order tensor analysis can be achieved by higher-order singular value decomposition (HOSVD) with *matrix unfolding*, which is a generalization of SVD [72] (See details in Appendix A 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 $\boldsymbol{A}_{(n)} = unfolding(\boldsymbol{\mathcal{A}}, n)$, is flattening $\mathcal{A}$ into a matrix whose column vectors are the mode-n vectors [72]. Therefore, the dimension of the unfolded matrix $\boldsymbol{A}_{(\boldsymbol{n})}$ is $N_n \times (N_1 \times N_2 \times \cdots N_{n-1} \times N_{n+1} \times \cdots N_r)$.

Each of the coefficient matrices $\boldsymbol{C}^{sv}$ can be represented as a coefficient vector $\boldsymbol{c}^{sv}$ by column stacking (stacking its columns above each other to form a vector). Therefore, $\boldsymbol{c}^{sv}$ is an $N_c = d \cdot N$ dimensional vector. All the coefficient vectors can then be arranged in an order-three gait coefficient tensor $\mathcal{C}$ with dimensionality $N_s \times N_v \times N_c$ corresponding to people shape styles, views, and, content basis, respectively. The coefficient tensor is then decomposed as

$$\boldsymbol{\mathcal{C}} = \tilde{\boldsymbol{\mathcal{D}}} \times_1 \tilde{\boldsymbol{S}} \times_2 \tilde{\boldsymbol{V}} \times_3 \tilde{\boldsymbol{F}}$$

where $\tilde{\boldsymbol{S}}$ is the mode-1 basis of $\boldsymbol{\mathcal{C}}$, which represents the orthogonal basis for the style space. Similarly, $\tilde{\boldsymbol{V}}$ is the mode-2 basis representing the orthogonal basis of the view space and $\tilde{\boldsymbol{F}}$ represents the basis for the mapping coefficient space. The dimensionality of these matrices are $N_s \times N_s$, $N_v \times N_v$, $N_c \times N_c$ for $\tilde{\boldsymbol{S}}$,$\tilde{\boldsymbol{V}}$ and $\tilde{\boldsymbol{F}}$ respectively. $\boldsymbol{\mathcal{D}}$ is a core tensor, with dimensionality $N_s \times N_v \times N_c$ which governs the interactions among 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 [73]. Final subspace representation is

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{D}} \times_1 \boldsymbol{S} \times_2 \boldsymbol{V} \times_3 \boldsymbol{F} \tag{4.23}$$

where the reduced dimensionality for $\boldsymbol{\mathcal{D}}$, $\boldsymbol{S}$, $\boldsymbol{V}$, and $\boldsymbol{F}$ are $n_s \times n_v \times n_c$, $N_s \times n_s$, $N_v \times n_v$, and $N_c \times n_c$ where n_s, n_v and n_c are the number of basis retained for each factor respectively. Using tensor multiplication we can obtain coefficient eigenmodes which is a new core tensor formed by $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{D}} \times_3 \boldsymbol{F}$ with dimension $n_s \times n_v \times N_c$.

Given this decomposition and given any n_s dimensional style vector $\boldsymbol{s}$ and any n_v dimensional view vector $\boldsymbol{v}$, we can generate coefficient matrix $\boldsymbol{C}^{sv}$ by unstacking the vector $\boldsymbol{c}^{sv}$ obtained by tensor product $\boldsymbol{c}^{sv} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{v}$. 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. 4.20. Therefore, the whole model for generating image $\boldsymbol{y}_t^{sv}$ can be expressed as

$$\boldsymbol{y}_t^{sv} = \textit{unstacking}(\boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{v}) \cdot \psi(\boldsymbol{x}_t)$$

This can be expressed abstractly also in the form of Eq. 4.7 by arranging the tensor $\boldsymbol{\mathcal{Z}}$ into a order-four tensor $\boldsymbol{\mathcal{A}}$ with dimensionality $d \times n_s \times n_v \times N$.

4.5 Solving for Multiple Factors

Given a 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 orthogonal factors as well as body configuration. We discriminate here between two cases: 1: *Input is a whole motion cycle.* 2: *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 orthogonal factors by aligning the input sequence manifold to the model conceptual manifold representation. For the second case, we introduce an iterative solution. Without lose of generality, and similar to the previous section, we will use the gait model in Eq. 4.18 as an example in this section, while solving for more factors are straight forward generalization.

Solving View and Style Given a Whole Sequence

Given a sequence of images representing a whole motion cycle, we can solve for the view parameter, $\boldsymbol{v}$, and shape style parameter, $\boldsymbol{s}$. First the sequence is embedded to a unit circle and aligned to the model as described in Sec. 4.4.1. Then, mapping coefficients $\boldsymbol{C}$ is learned from the aligned embedding to the input. Given such coefficients, we need to find the optimal $\boldsymbol{s}$ and $\boldsymbol{v}$ factors which can generate such coefficients given the learned model. i.e., we need to find $\boldsymbol{s}$ and $\boldsymbol{v}$ which minimizes the error

$$E(\boldsymbol{s}, \boldsymbol{v}) = \|\boldsymbol{c} - \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{v}\| \tag{4.24}$$

where $\boldsymbol{c}$ is the column stacking of $\boldsymbol{C}$. If the style vector, $\boldsymbol{s}$ is known we can obtain a closed form solution for $\boldsymbol{v}$. This can be achieved by evaluating the product $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s}$ to obtain tensor $\boldsymbol{\mathcal{G}}$. Solution for $\boldsymbol{c}$ can be obtained by solving the system $\boldsymbol{c} = \boldsymbol{\mathcal{G}} \times_2 \boldsymbol{v}$ for $\boldsymbol{v}$ which can be written as a typical linear system by unfolding $\boldsymbol{\mathcal{G}}$ as a matrix. Therefore, estimate of $\boldsymbol{v}$ can be obtained by

$$\boldsymbol{v} = (\boldsymbol{G}_2)^{\dagger}\boldsymbol{c} \tag{4.25}$$

where $\boldsymbol{G}_2$ is the matrix obtained by mode-2 unfolding of $\boldsymbol{\mathcal{G}}$ and † denotes the pseudo-inverse using SVD. Similarly we can analytically solve for $\boldsymbol{s}$ if the view, $\boldsymbol{v}$, is known by forming a tensor $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{Z}} \times_2 \boldsymbol{v}$ and therefore

$$\boldsymbol{s} = (\boldsymbol{H}_1)^{\dagger}\boldsymbol{c} \tag{4.26}$$

where $\boldsymbol{H}_1$ is the matrix obtained by mode-1 unfolding of $\boldsymbol{\mathcal{H}}$

Iterative estimation of $\boldsymbol{v}$ and $\boldsymbol{s}$ using Eq. 4.25 and Eq. 4.26 would lead to a local minima for the error in Eq. 4.24. We start with a mean style estimate $\tilde{s}$ since we don't know styles at the beginning. Since the view classes are discrete, we can find the closest view class and use it to estimate $\boldsymbol{s}$.

Solving for Body Configuration, View and Style from a Single Image

In this case the input is a single image and it is desired to estimate body configuration and each of the decomposable factors. For the gait case, given an input image $\boldsymbol{y}$, we need to estimate body configuration $\boldsymbol{x}$, view $\boldsymbol{v}$, and person shape style $\boldsymbol{s}$ which minimize the reconstruction error $E(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{s})$

$$E(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{s}) = || \boldsymbol{y} - \boldsymbol{\mathcal{A}} \times \boldsymbol{v} \times \boldsymbol{s} \times \psi(\boldsymbol{x}) || \tag{4.27}$$

We can instead use a robust error metric and in both cases we end up with a nonlinear optimization problem.

We assume optimal style can be written as a linear combination of style classes in the training data. i.e., we need to solve for linear regression weights α such that $\boldsymbol{s} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$ where each $\boldsymbol{s}^k$ is a mean of one of K_s style classes in the training data. Similarly for the view, we need to solve for weights β such that $\boldsymbol{v} = \sum_{k=1}^{K_v} \beta_k \boldsymbol{v}^k$ where each $\boldsymbol{v}^k$ is a mean of one of K_v view classes. If the style and view factors are known, then Eq. 4.27 reduced to a nonlinear 1-dimensional search problem for body configuration $\boldsymbol{x}$ on the embedded manifold representation that minimizes the error. On the other hand, if the body configuration and style factor are known, we can obtain view conditional class probabilities $p(\boldsymbol{v}^k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$ which is proportional to observation likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}^k)$. Such likelihood can be estimated assuming a Gaussian density centered around $\boldsymbol{\mathcal{A}} \times \boldsymbol{v}^k \times \boldsymbol{s} \times \psi(\boldsymbol{x})$, i.e.,

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}^k) \approx \mathcal{N}(\boldsymbol{\mathcal{A}} \times \boldsymbol{v}^k \times \boldsymbol{s} \times \psi(\boldsymbol{x}), \Sigma^{\boldsymbol{v}^k}).$$

Given view class probabilities, we can set the weights to $\beta_k = p(\boldsymbol{v}^k \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$. Similarly, if the body configuration and view factor are known, we can obtain style weights by evaluating image likelihood given each style class $\boldsymbol{s}^k$ assuming a Gaussian density centered at $\boldsymbol{\mathcal{A}} \times \boldsymbol{v} \times \boldsymbol{s}^k \times \psi(\boldsymbol{x})$.

This setting favors an iterative procedures for solving for $\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{s}$. However, wrong estimation of any of the factors would lead to wrong estimation of the others and leads to a local minima. For example wrong estimation of the view factor would lead to a totally wrong estimate of body configuration and therefore wrong estimate for shape style. To avoid this we use a deterministic annealing like procedure where at the beginning the view weights and style weights are forced to be close to uniform weights to avoid hard decisions about view and style classes. The weights are gradually become discriminative thereafter. To achieve this, we use a variable view and style class variances which are uniform to all classes and are defined as $\Sigma^v = T_v \sigma_v^2 \boldsymbol{I}$ and $\Sigma^s = T_s \sigma_s^2 \boldsymbol{I}$ respectively. The parameters T_v and T_s start with large values and are gradually reduced and in each step and a new body configuration estimate is computed. We summarize the solution framework in Fig. 4.3

```
Input: image $\boldsymbol{y}$, view class means $\boldsymbol{v}^k$, style class means $\boldsymbol{s}^k$, core tensor $\boldsymbol{\mathcal{A}}$
Initialization :
        • initialize $T_v$ and $T_s$
        • initialize $\alpha_k$ and $\beta_k$ to uniform weights
        • Compute initial $\boldsymbol{s} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$
        • Compute initial $\boldsymbol{v} = \sum_{k=1}^{K_v} \beta_k \boldsymbol{v}^k$
Iterate :
        • Compute coefficient $\boldsymbol{C} = \boldsymbol{\mathcal{A}} \times \boldsymbol{s} \times \boldsymbol{v}$
        • Estimate body configuration: 1-D search for $\boldsymbol{x}$ that minimizes $E(\boldsymbol{x}) = ||\boldsymbol{y} - \boldsymbol{C}\psi(\boldsymbol{x})||$
        • Estimate new view factor
            – Compute $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}^k)$
            – Update view weights $\beta_k = p(\boldsymbol{v}^k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$
            – Estimate new view factor as $\boldsymbol{v} = \sum_{k=1}^{K_v} \beta_k \boldsymbol{v}^k$
        • Update coefficient $\boldsymbol{C} = \boldsymbol{\mathcal{A}} \times \boldsymbol{s} \times \boldsymbol{v}$
        • Estimate body configuration: 1-D search for $\boldsymbol{x}$ that minimizes $E(\boldsymbol{x}) = ||\boldsymbol{y} - \boldsymbol{C}\psi(\boldsymbol{x})||$
        • Estimate new style factor
            – Compute $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{s}^k, \boldsymbol{v})$
            – Update style weights $\alpha_k = p(\boldsymbol{s}^k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{v})$
            – Estimate new style factor as $\boldsymbol{s} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$
        • reduce $T_v, T_s$
```

Figure 4.3: Iterative estimation of style factors

One important aspect that needs to be mentioned for the case of gait is that there is a high similarity between silhouette shapes in each of the half cycles for certain views. In fact, if orthographic projection is used, side view silhouettes will look identical in both halves of the walking cycle. But since perspective imaging is actually used, there is slight differences in silhouette shapes between the two half cycles which are enough to discriminate body configuration throughout the cycle. However, such similarity can cause a confusion in estimating $\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}$. This motivates a modification of the above algorithm for the spacial case of gait where we use dual hypotheses for body configuration and view and style factors. At initialization we solve for body configuration $\boldsymbol{x}$ given the mean style and mean view factors then we initializes dual body configuration hypotheses as $\boldsymbol{x}$ and its antipodal point on the circle which we call $\tilde{\boldsymbol{x}}$. The iterations above proceed with two sets of estimates $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v})$ and $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{v}})$. The two sets typically either converge to the same solution or they diverge to two antipodal body configurations where one of them will lead to less error.

4.6 Experimental Results

4.6.1 Dynamic Shape: Generative Model for Gait

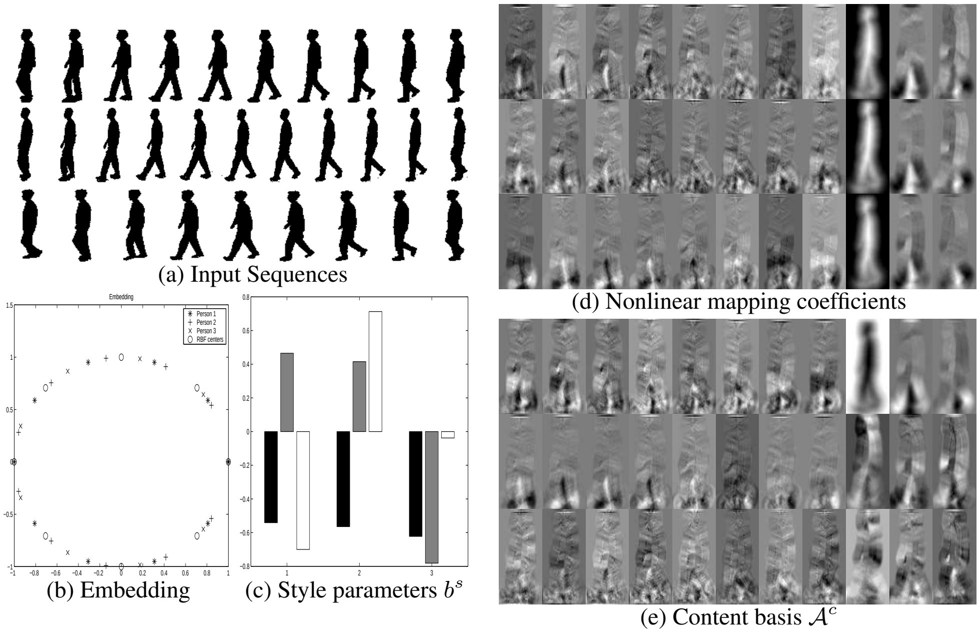
In this section we show an example of learning the nonlinear manifold of gait as an example of a dynamic shape. We used CMU Mobo gait data set [48] which contains walking people from multiple synchronized views. The CMU Mobo gait data set contains 25 people, about 8 to 11 walking cycles captured from six different view points. Each subject walks on treadmill to capture gait sequences with consistent view using fixed cameras.

Gait Style and Content Analysis

We used side view gait sequences from CMU Mobo gait dataset for gait style and content analysis from a single view.

Dynamic Shape Interpolation In this experiment we use three people's silhouettes during a half walking cycle to separate the style (person shape) from the content (body pose). The input is three sequences containing only 10, 11, 9 frames respectively. The input silhouettes are shown in Fig. 4.4 (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 using LLE, we arbitrary embed the data points on a 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. 4.4 (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. 4.4 using TPS kernels. Fig. 4.4 (d) shows the RBF coefficients for the three people (one in each row) where the last three columns are the polynomial coefficients. Fig. 4.4 (c) shows the style coefficients for the three people and Fig. 4.4 (e) show 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. 4.5 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 at new styles. This is shown in Fig. 4.5 where intermediate styles and intermediate contents were used.



(a) Input Sequences

(b) Embedding

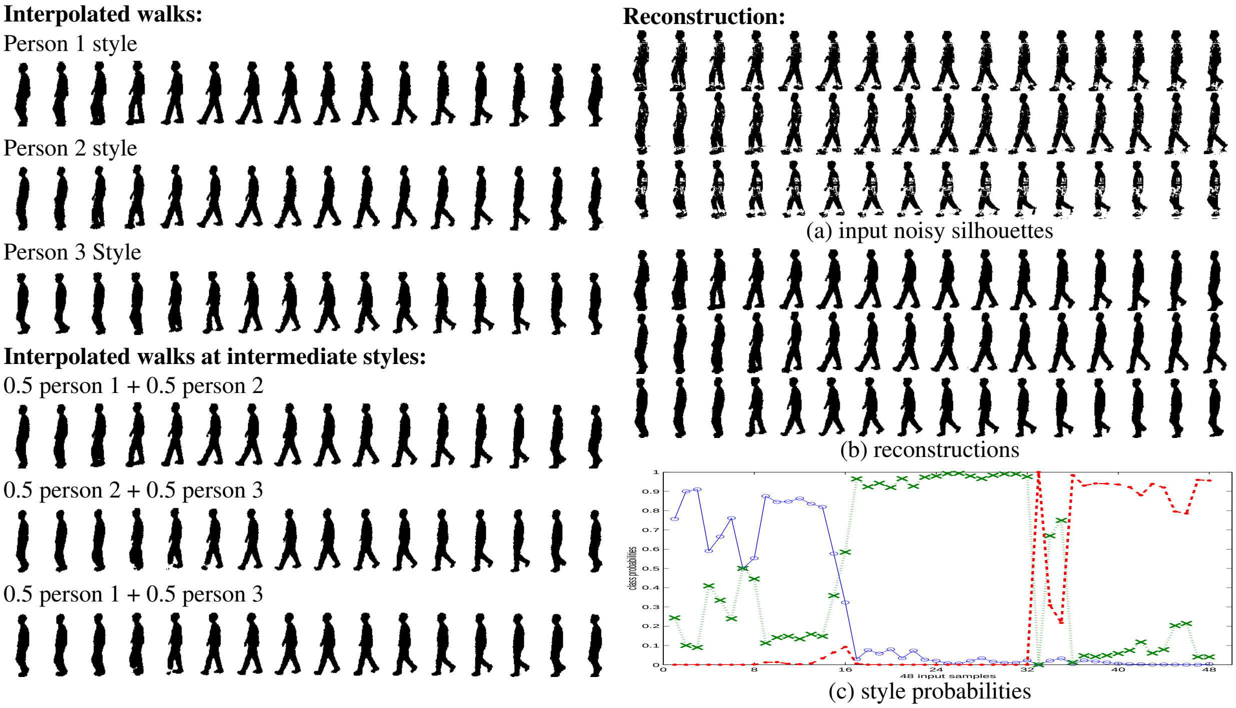
(c) Style parameters b^s

(d) Nonlinear mapping coefficients

(e) Content basis $\mathcal{A}^c$

Figure 4.4: Learning style and content for a gait example

Style-Preserving Pose-Preserving Reconstruction: We can use the learned model to reconstruct noisy and corrupted input instances in a way that preserve both the body pose and the person style. Given an input silhouette we solve for both the embedding coordinate and the



(a) input noisy silhouettes

(b) reconstructions

(c) style probabilities

Figure 4.5: 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.

style as was described in Sec. 4.3.3 and use the model to reconstruct a corrected silhouette given the recovered pose and person parameters. Fig. 4.5 shows such reconstruction where we used 48 noisy input silhouettes from CMU Mobogait database were used (16 for each person shown at each row). The resulting people's probabilities are shown in Fig. 4.5 (c) and the resulting reconstructions are shown in Fig. 4.5 (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 the person 3 and therefore the reconstruction preserved the second person style. Fig. 4.6 shows another reconstruction example where the learned model was used to reconstruct corrupted inputs for person 3. The reconstruction preserve the person style as well as the body pose.

Manifold Embedding and Style Classification:

In this experiment we used five sequences for five different people each containing about 300 frames which are noisy. The learned manifolds are shown in Fig. 4.8 (a) which shows a different manifold for each person. The learned unified manifold is also shown in Fig. 4.8

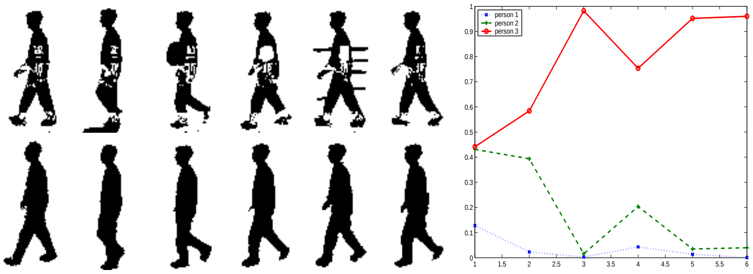


Figure 4.6: Pose and style preserving reconstruction. Right: style probabilities for each input

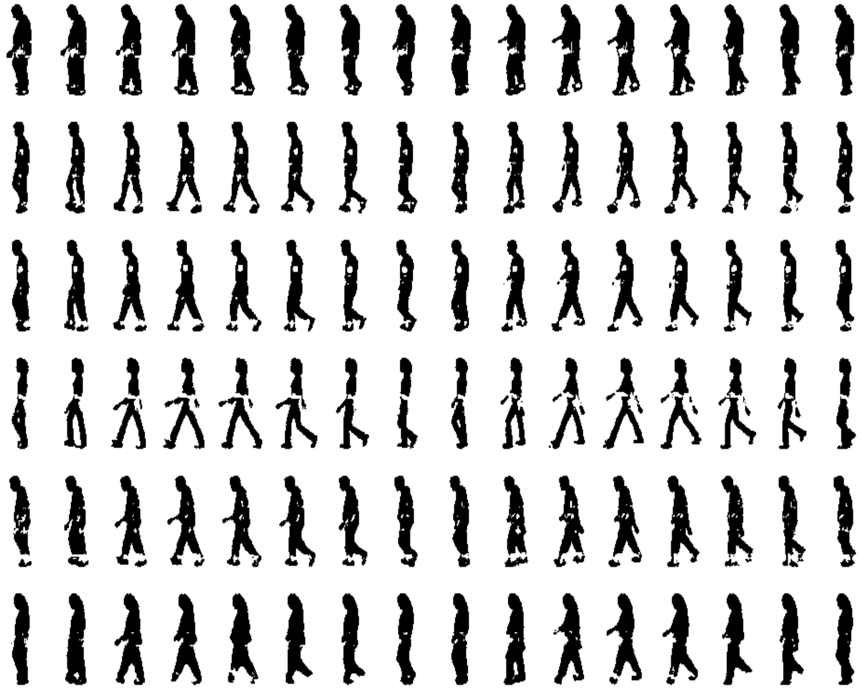
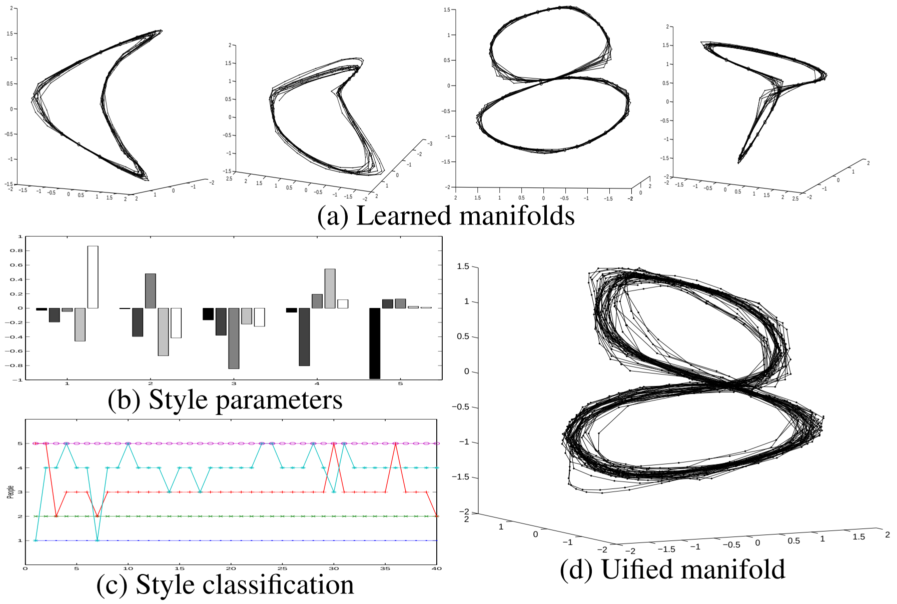


Figure 4.7: Interpolated walks. Last row is interpolated walk at intermediate style between row 1 and 4.

(d). Fig. 4.7 shows interpolate 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 result is shown in the figure with correct classification rate of 92%. We also used the learned model to interpolate walks in new styles. The last row in the figure shows interpolation between person 1 and person 4.

Multiple Factors Model for Gait

For learning decomposable dynamic shape models with multiple views in addition to style difference and body configuration change, we selected five people, five cycles each from four



(a) Learned manifolds

(b) Style parameters

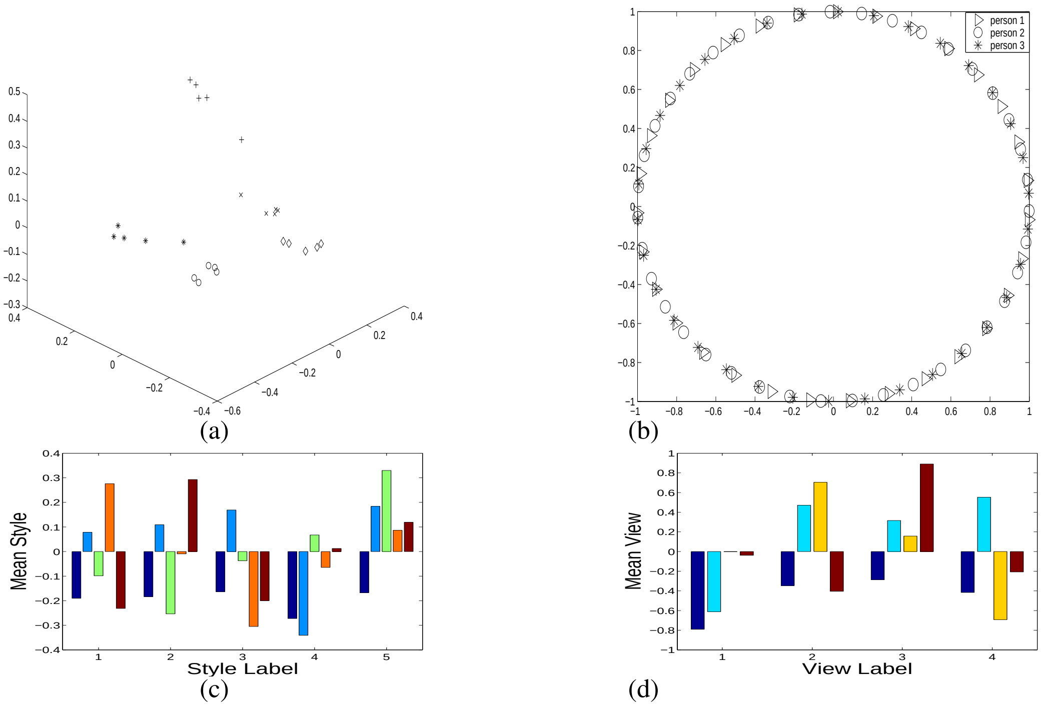
(c) Style classification

(d) Uified manifold

Figure 4.8: Style estimation

different views. i.e., the total number of cycles for training is 100 = 5 people × 5 cycles × 4 views. Note that the number of frames in each cycle is different within the same person's cycles as well as in different people. Fig. 4.1 show examples of the sequences with different views (only half cycles are shown in the figure).

We learned a generative model with three decomposable factors from the collected 100 cycle sequences as described in Sec. 4.4.2. Images are normalized to 60 × 100 (width × height) i.e., $d = 6000$. Each cycle is considered to be a style by itself, i.e., there are 25 styles and 4 views. Therefore, $N_s = 25$, $N_v = 4$ in the collected data. 18 equidistance points on the unit circle are used to obtain the nonlinear mapping defined in Eq. 4.11, i.e., $N_c = 6000 \times 18$. After coefficient decomposition and dimensionality reduction as in Eq. 7.6 the dimension for $\mathcal{A}, S, V, F$ are $5 \times 4 \times 120$, 25×5, 4×4, $(18 \times 6000) \times 120$ respectively. Fig. 4.9 (b) shows an example of unit circle embedding of three cycles after alignment of cycles using Eq. 4.22. Fig. 4.9 (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 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. 4.9 (c) shows the mean style



(a) (b) (c) (d)

Figure 4.9: 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 cluster. (d) View vectors.

vector for each of the five clusters. Fig. 4.10 shows the four view vectors.

Gait Pose, Style, and View Estimation: In this experiment, we used the learned model given the training data described above to evaluate the recovery of body configuration, view, and person shape style given test data of the same people in the training but with different cycles, which are not used in the training. We used two new cycles for each of the five people from the four views, i.e., 40 cycles with a total of 1344 frames in all the test sequences. If we use a whole cycle for recovery of view and person style parameter as described in 4.5, 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 single frames for recovery, as described in Sec. 4.5, we get 7 frame errors among 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 correct and estimated embedding is more than $\pi/8$, which is about 2 to 4 frame difference in the original sequence.

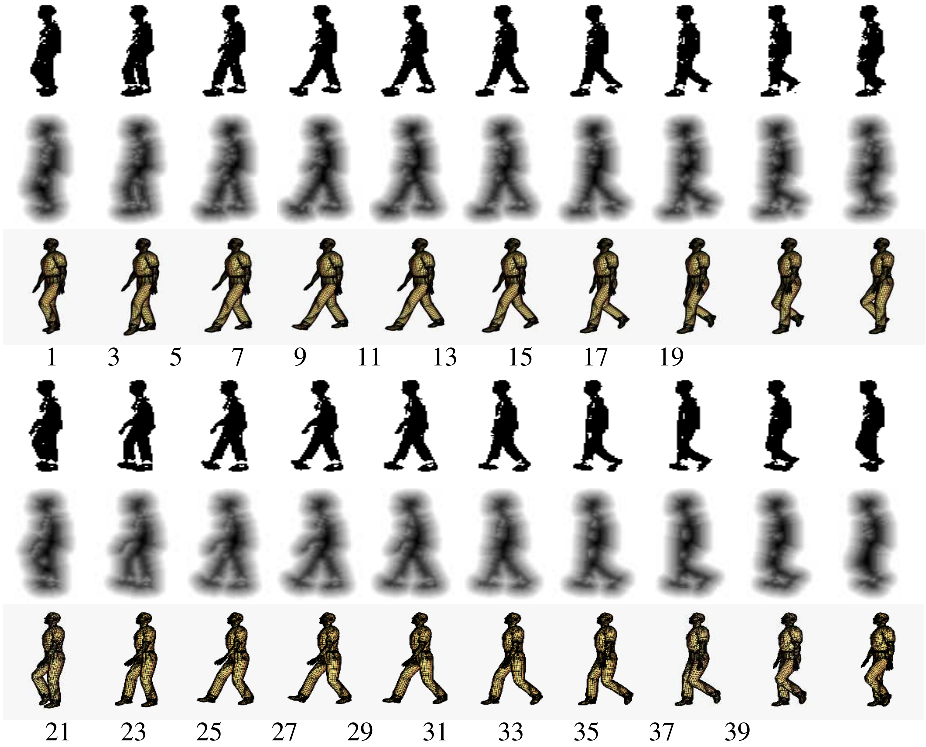


Figure 4.10: Example pose recovery. From top to bottom: input shapes, implicit function, recovered 3D pose.

Fig. 4.10 shows example of using the model to recover the pose, view and style. The figure shows samples of a 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. 4.20. Fig. 4.11 (a),(b) shows the recovered style weights (class probabilities) and view weights respectively for each frame of the cycle which shows correct person and view classification. Fig. 4.11 (c) visualizes the progress of the error, style weights, 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 to the correct style and view as the error is reduced and the correct body configuration is recovered.

Generalization to New Subject: In this experiment we used the learned model to evaluate the recovery of body configuration and view given test data of people which have not seen before in the training. We used 8 people sequences, 2 cycles each, from 4 views where none

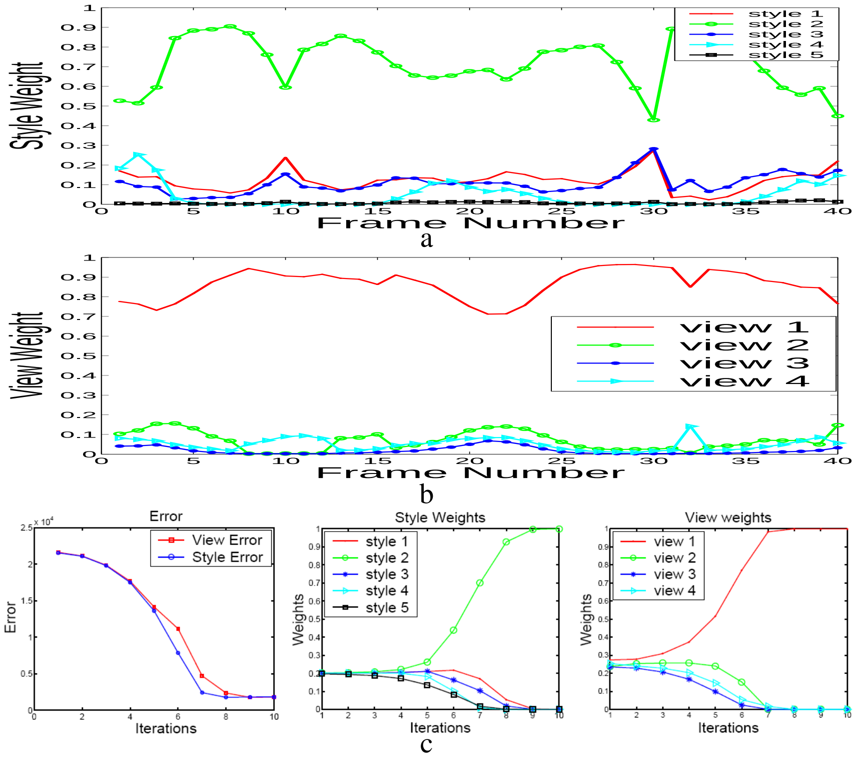


Figure 4.11: Estimated weights during a cycle. (a) Style weights. (b) View weights. (c) Iterative style and view estimations for each frame. Left: error. Center: style weights. Right: view weights

of these people 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 Sec. 4.5. We obtained 111 errors in the recovery of the body configuration, i.e., body configuration accuracy is 95.52%. For view estimation we get 7 frame errors, i.e., view estimation accuracy 99.72%. This result shows that the model generalizes and we can recover the view and body configuration with very high accuracy for unseen people. Fig. 4.12 shows examples recovery of the 3D pose and view class for different people non of them was seen in training. More examples can be seen in the attached video clips.

4.6.2 Dynamic Appearance: Generative Model for Facial Expressions

We used CMU-AMP facial expression database where each subject has 75 frames of varying facial expressions.

Learning Smile Manifold and Style Analysis: In this experiment the proposed model was

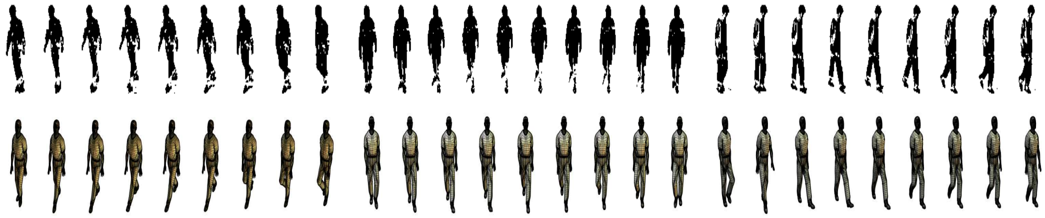


Figure 4.12: Examples of pose recovery and view classification for four people.

used to learn the manifold of a smile and separate the appearance (style) for 4 people[1]. The input sequences contain 27,31,29,27 frames respectively for the smile motion. All the input sequences were temporally scaled from 0 to 1 then LLE were used to obtain a one-dimensional embedding of the manifolds and a unified embedding is obtained as was described in Sec. 4.2. The model was fitted using 8 equally spaced RBF centers along the mean manifold. The first four rows of Fig. 4.13 show interpolation of 10 intermediate faces at each of the learned styles. As can be noticed, the model is able to correctly interpolate the facial motion of the smile for the four people. It is hard to prove in this case that the model is actually interpolating new intermediate faces but we can easily show interpolating smiles at new styles. This is shown in the last three rows where the model is used to render smiles at intermediate styles.

Multiple Factors: Modeling Multiple Facial Expressions and Multiple People Appearance: We used the model to learn facial expression manifolds for different people. We chose four people with three expressions each (smile, anger, surprise) where corresponding frames are manually segmented from the whole sequence for training. The resulting training set contained 12 sequences of different lengths. Fig. 4.14 shows the training data. All sequences are embedded to unit circles and aligned. A model in the form of Eq. 4.6 is fitted to the data where we decompose two factors: person facial appearance style factor and expression factor, besides the body configuration which is nonlinearly embedded on a unit circle. Fig. 4.14 shows the resulting person style vectors and expression vectors.

We used the learned model to recognize facial expression, and person identity at each frame of the whole sequence. Fig. 4.15-Left shows an example of a whole sequence and the different expression probabilities obtained on a frame per frame basis using the algorithm described in Sec. 4.5. The figure also shows the final expression recognition after thresholds along manual

[1] The images are from the CMU facial expression data set

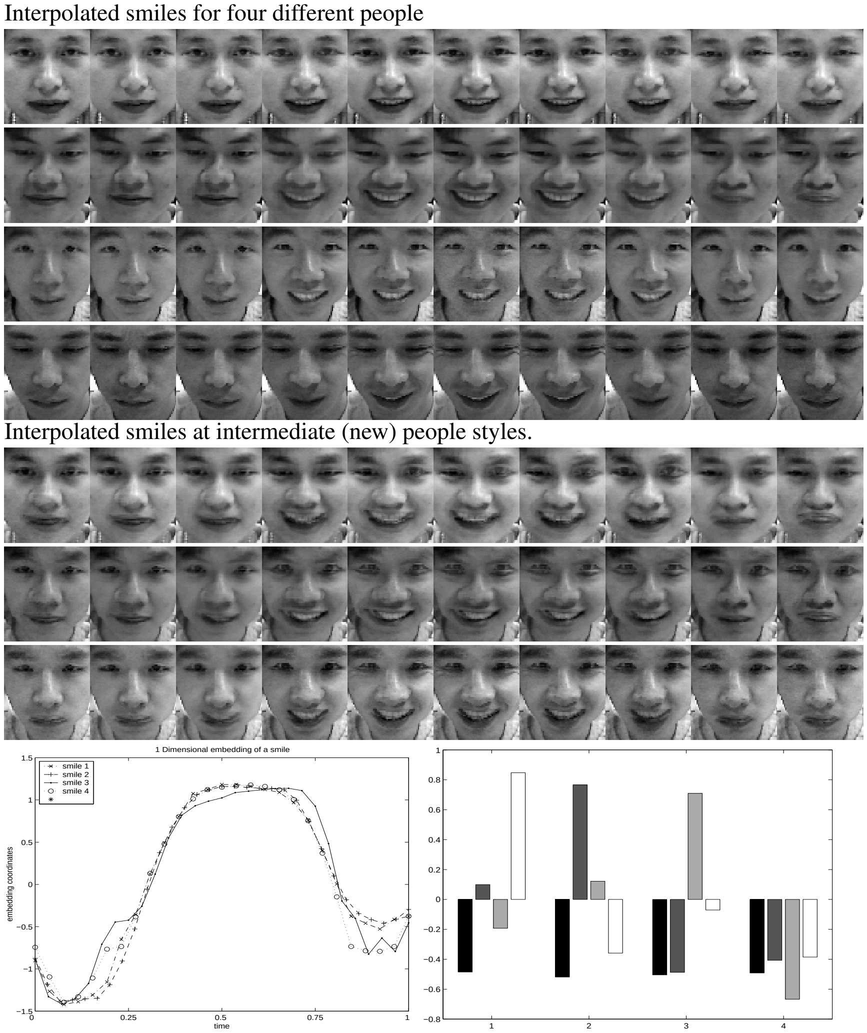


Figure 4.13: Learning a smile manifold. bottom: manifold embedding and style parameters

expression labeling. We used the learned model to recognize facial expressions for sequences of people not used in the training. Fig. 4.15-Right shows an example of a sequence of a person not used in the training. The model can successfully generalizes and recognize the three learned expression for this new subject.

4.7 Summary

We introduced a framework for separating style and content on manifolds representing dynamic objects. The framework is based on decomposing the style parameters in the space of nonlinear functions that maps between a learned unified nonlinear embedding of multiple content manifolds and the visual input space. The framework yields an unsupervised procedure that handles

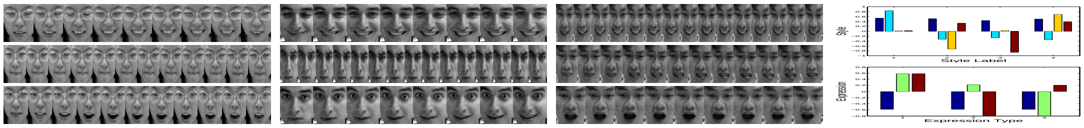


Figure 4.14: Left: Example of training sequences for facial expression. Three people (out of four) and three expressions. Right-top:Style vectors for each person. Right-bottom: Expression vectors

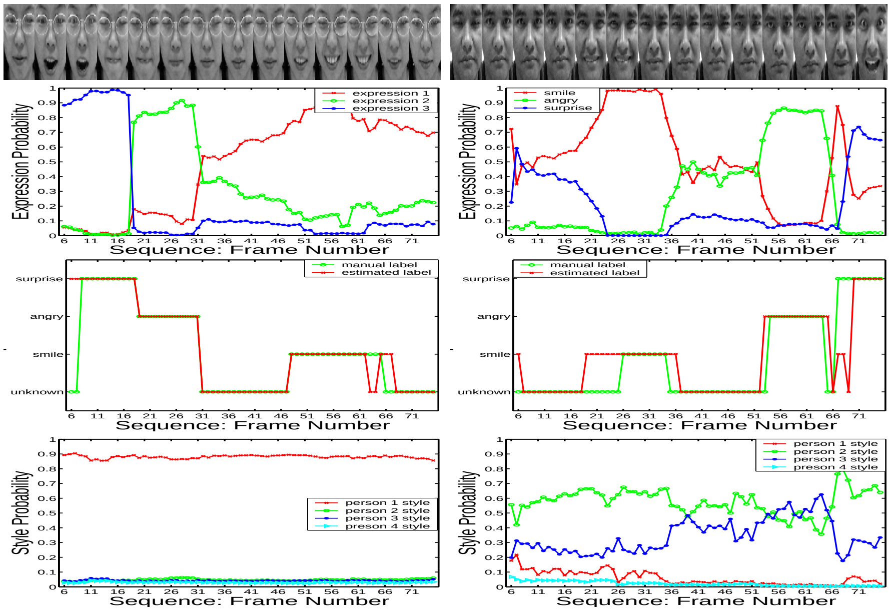


Figure 4.15: Expression recognition and appearance style: Left: Person with known style. Right: Unknown Person. From top to bottom: Samples of the input sequences; Expression probabilities; Expression classification; Style probabilities

dynamic, nonlinear manifolds. It also improves on past work in nonlinear dimensionality reduction by being able to handle multiple manifolds. The proposed framework was shown to be able to separate

various factors such as body configuration, view, and shape style. Since the framework is generative, it fits well in a Bayesian tracking framework and it provides separate low dimensional representations for each of the modeled factors. Moreover, a dynamic model for configuration is well defined since it is constrained to the 1D manifold representation. The framework also provides a way to initialize a tracker by inferring about body configuration, view point, body shape style from a single or a sequence of images.

Chapter 5

Style Adaptive Contour Tracking Using Decomposable Generative Models

Characteristics of the shape deformation in human motion contain rich information and can be useful for human identification, gender classification, 3D pose reconstruction and so on. In this paper we introduce a new framework for dynamic contour tracking of human motion using an explicit modeling of the motion manifold and learning a decomposable generative model. We use nonlinear dimensionality reduction to embed the motion manifold in a low dimensional configuration space utilizing the constraints imposed by the human motion. Given such embedding, we learn an explicit representation of the manifold, which reduces the problem to a one-dimensional tracking problem and also facilitates linear dynamics on the manifold. Person-dependent global shape deformations are modeled using a nonlinear generative model with kinematic manifold embedding and kernel mapping. A person shape style factor as well as geometric transformation and body pose are estimated within a Bayesian framework using the generative model of global shape deformation. Experimental results show person-dependent synthesis of global shape deformation, gait recognition from extracted silhouettes using person shape style parameters, and simultaneous gait contour tracking and recognition from image edges.

5.1 Overview: Contour Tracking

Vision-based human motion tracking and analysis systems have promising potentials for many applications such as visual surveillance in public area, activity recognition, and sport analysis. Human motion involves not only geometric transformations but also deformations in shape and appearance. Characteristics of the shape deformation in a person motion contain rich information such as body configuration, person identity, gender information [34], and even emotional states of the person. Gait recognition has become attractive for surveillance and for security in public areas [32, 56, 9] as it is easily observable and difficult to disguise than other biometrics.

Gait involves spatiotemporal deformations in shape and appearance. Such spatiotemporal shape deformation are investigated in many appearance-based gait recognition systems [9, 86, 130, 65, 80, 142, 76] (See detailed related work for gait recognition in Sec. 2.1.3). On the other hand, there have been a lot of work on contour tracking from cluttered environment, without the need for background subtraction, such as active shape models (ASM) [30], active contours [58], and exemplar-based tracking [131]. Spatiotemporal models are also used for contour tracking [5]. However, it is difficult to achieve tracking of dynamic contour that is accurate enough to distinguish individual differences from articulated human motion. There are no spatiotemporal models for contour tracking to describe person-specific variations of shape for gait recognition.

Modeling dynamics of shape and appearance is essential for tracking human motion. The observed human shape and appearance in video sequences goes through complicated global nonlinear deformation between frames. If we consider the global shape, there are two factors affecting the shape of the body contour through the motion: *global dynamics factor* and *person shape style factor*. The dynamic factor is constrained because of dynamics of the motion and the physical characteristics of human body configuration [98, 17]. The person shape style is time-invariant factor characterizing distinguishable features in each person shape depending on body built(big, small, short, tall, etc.). These two factors can summarize rich characteristics of human motion and identity.

Our objective is to achieve trackers that can track global deformation in contours and can adapt to different people shapes automatically. There are several challenges to achieve this goal. First, modeling the human body shape space is hard, considering both the dynamics and the shape style. Such shapes lie on a nonlinear manifold. Also, in some cases there are topological changes in contour shapes through motion which makes establishing correspondences between contour points unfeasible. Second, modeling dynamics of global shape is important for tracking. Can we learn a dynamic model for body configuration that is low in dimensionality and exhibits linear dynamics? For certain classes of motion like gait, facial expression and gestures, the deformation might lie on a low dimensional manifold if we consider a single person.

Nonlinear manifold learning can be used to find intrinsic body configuration space [143, 37].

We utilized generative models for simultaneous tracking and recognition of gait. As the generative model is represented by a configuration state and a shape style state, the shape style state is a compact representation of variations in shape contours independent of body pose (the configuration state). We use the estimated style for gait recognition. When the extracted silhouette is provided (e.g. using background subtraction), we can directly estimate the contour style state and recognize gait based on the estimated contour style parameters. On the other hand, if extracted silhouette is not available, we use contour tracking where the tracing problem is formulated as estimation of body configuration state as well as contour style state using Bayesian framework. Style estimation gradually get discriminative using deterministic annealing like procedure in order to estimate contour style state, which can be high dimensional, robustly without trapping to local minima. Experimental results using University of Southampton gait database [118] shows potential for simultaneous gait recognition and contour tracking.

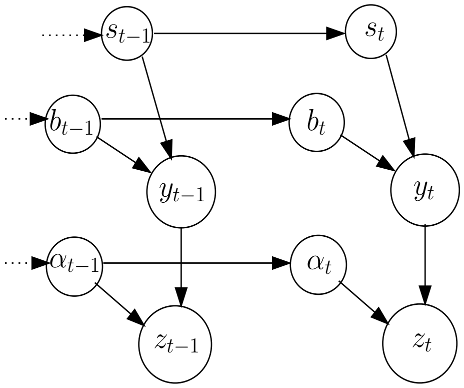


Figure 5.1: Graphic model for decomposed generative model

5.2 Framework: Tracking Using Decomposable Generative Models

We can think of the shape of a dynamic object as instances driven from a generative model. Let $\boldsymbol{z}_t \in R^d$ be the shape of the object at time instance t represented as a point in a d-dimensional space. This instance of the shape is driven from a model in the form

$$\boldsymbol{z}_t = T_{\boldsymbol{\alpha}_t}\gamma(\boldsymbol{b}_t, \boldsymbol{s}_t; \boldsymbol{a}), \tag{5.1}$$

where the $\gamma(\cdot)$ is a nonlinear mapping function that maps from a representation of the body configuration $\boldsymbol{b}_t$ and a representation of the shape space $\boldsymbol{s}_t$ into the observation space given mapping parameters denoted by $\boldsymbol{a}$. $T_{\boldsymbol{\alpha}_t}$ represents a geometric transformation on the shape instance.

Fig. 5.1 shows a graphical model illustrating the relation between these variables where $\boldsymbol{y}_t$ is a contour instance generated from model given body configuration $\boldsymbol{b}_t$ and shape style $\boldsymbol{s}_t$ and transformed in the image space through $T_{\boldsymbol{\alpha}_t}$ to form the observed contour. The mapping $\gamma(\boldsymbol{b}_t, \boldsymbol{s}_t; \boldsymbol{a})$ is a nonlinear mapping from the body configuration state $\boldsymbol{b}_t$ and the shape state $\boldsymbol{s}_t$ in the form

$$\boldsymbol{y}_t = \mathcal{A} \times \boldsymbol{s}_t \times \psi(\boldsymbol{b}_t), \tag{5.2}$$

where $\psi(\boldsymbol{b}_t)$ is a kernel induced space, $\mathcal{A}$ is a third order tensor and $\times$ is appropriate tensor product as well be defined in Sec. 5.3.2. The tensor $\mathcal{A}$ characterizes the model parameters and controls the correlation between the configuration state and the shape state.

Given this generative model, we can fully describe observation instance $\boldsymbol{z}_t$ by state parameters $\boldsymbol{\alpha}_t$,$\boldsymbol{b}_t$, and $\boldsymbol{s}_t$. The challenges to achieve learning and tracking using such model include: How to represent the body configuration in a low dimensional space? How to represent the shape space? How to estimate the parameters? How to deal with heterogenous state representation in tracking? We use a low dimensional embedded representation of the motion manifold to represent the body configuration. In this paper, since we focus on gait tracking, the dimensionality of the body configuration space reduces to an one-dimensional space and we show that the resulting dynamics using such representation is a constant speed linear dynamics.

The shape style is represented using a linear combination of learned shape style classes. The shape variable $\boldsymbol{s}_t$ characterizes the person shape style in a way independent from the configuration and specific to the person being tracked. Therefore, ideally, the shape style variable should be time invariant. However, since in tracking, the person shape style is unknown, we need to deal with it as a stochastic variable that changes with time in a way that allows the tracker to adapt to the person shape. Once, the person is tracked for few frames, the shape style is determined and need to be stabilized to be a time-invariant factor. That motivates a deterministic annealing-like procedure that we introduce for the estimation of the shape style variable.

The tracking problem is then an inference problem where at time t we need to infer the body configuration representation $\boldsymbol{b}_t$ and the person specific style parameter $\boldsymbol{s}_t$ and the geometric transformation $T_{\boldsymbol{\alpha}_t}$ given the observation $\boldsymbol{z}_t$. The Bayesian tracking framework enables a recursive update of the posterior $P(\boldsymbol{x}_t|\boldsymbol{z}^t)$ over the object state $\boldsymbol{x}_t$ given all observation $Z^t = z_1, z_2, .., z_t$ up to time t:

$$P(x_t|Z^t) \propto P(z_t|x_t) \int_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}|Z^{t-1}). \tag{5.3}$$

Observation $\boldsymbol{z}_t$ is the captured image instance at time t. The state $\boldsymbol{x}_t$ is $[\boldsymbol{\alpha}_t, \boldsymbol{b}_t, \boldsymbol{s}_t]$, which uniquely describes the state of the tracking object, is decomposed into three sub-states $\boldsymbol{\alpha}_t, \boldsymbol{b}_t, \boldsymbol{s}_t$. These three random variables are conceptually independent since we can combine any body configuration with any person shape style with any geometrical transformation to synthesize a new contour. However, they are dependent given the observation $\boldsymbol{z}_t$. It is hard to estimate joint posterior distribution $P(\boldsymbol{\alpha}_t, \boldsymbol{b}_t, \boldsymbol{s}_t|\boldsymbol{z}_t)$ for its high dimensionality. The decomposable feature of our generative model enables us to estimate each state by its marginal density distribution $P(\boldsymbol{\alpha}_t|\boldsymbol{Z}^t)$, $P(\boldsymbol{b}_t|\boldsymbol{Z}^t)$, and $P(\boldsymbol{s}_t|\boldsymbol{Z}^t)$. We approximate marginal density estimate of each state variable along representative values of the other state variables. For example, in order to estimate marginal density of $P(\boldsymbol{b}_t|\boldsymbol{Z}^t)$, we estimate $P(\boldsymbol{b}_t|\boldsymbol{\alpha}_t^*, \boldsymbol{s}_t^*, \boldsymbol{Z}^t)$, where $\boldsymbol{\alpha}_t^*, \boldsymbol{s}_t^*$ are representative values such as maximum posteriori estimates.

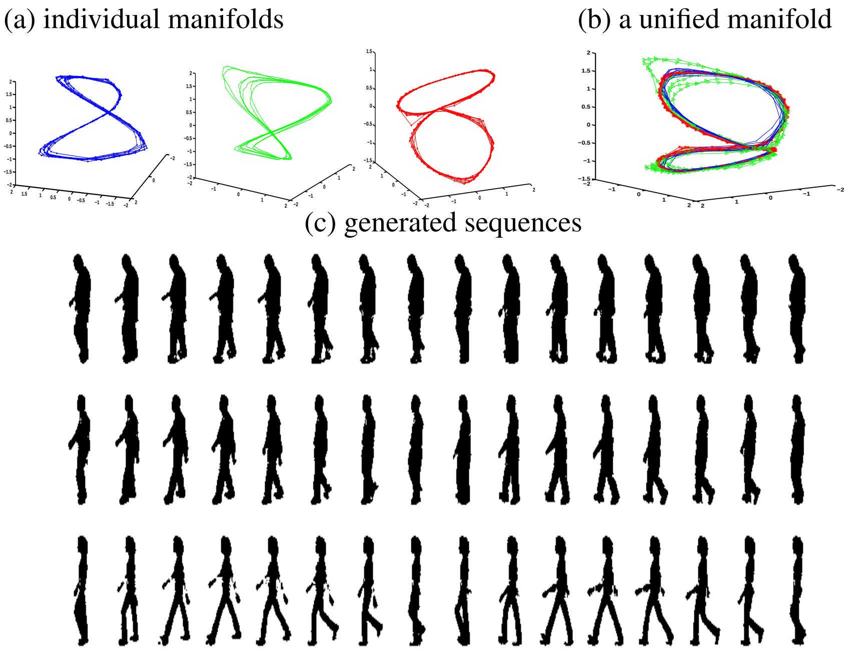
5.3 Learning Style Adaptive Shape Models with Body Configuration Manifold Embedding

Our objective is to establish a generative model for the shape in the form of Eq. 5.1 where the intrinsic body configuration is decoupled from the shape style. The generative model consists of three components: embedded body configuration ($\boldsymbol{b}_t$), factorized style ($\boldsymbol{s}_t$), and geometric transformation ($\boldsymbol{\alpha}_t$. In order to model body configuration in a low dimensional space independent of shape style variability. We introduce two alternatives to achieve this goal: visual manifold embedding, and kinematics manifold embedding.

5.3.1 Modeling Body Configuration Using Manifold Embedding

Given training sequences of different people performing the same motion (gait in our case), we propose two approaches to model body configuration manifold invariant of shape variability. One is driven by finding a unified representation from the different visual manifolds of each individual. The other is driven by using an invariant kinematic manifold from motion captured data, where we can model purely body configuration manifold regardless to changes of visual input.

Embedding Visual Manifolds


(a) individual manifolds

(b) a unified manifold

(c) generated sequences

Figure 5.2: Individual manifolds and their unified manifold

We apply Locally Linear Embedding (LLE) [111] to find low dimensional representation of body configuration for each person manifold. As a result of nonlinear dimensionality reduction, an embedding of the visual gait manifold can be obtained in a low dimensional Euclidean space. Fig. 5.2-a shows low dimensional representation of side-view walking sequences for different people. Generally, the walking cycle evolves along a closed curve in the embedded space, i.e., only one degree of freedom controls the walking cycle which corresponds to the constrained body pose as a function of the time. Such manifold can be used as intrinsic representation of the body configuration. The use of nonlinear manifold embedding to obtain intrinsic representation for tracking was previously reported in [143]. In [36], it was shown that for gait, a three-dimensional embedded space is enough to represent the gait where all body postures are distinguishable through the walking cycle.

Body configuration manifold is parameterized using a spline fitted to the embedded manifold representation. First, cycles are detected given an origin point on the manifold by computing geodesics along the manifold. Second, a mean-manifold for each person is obtained by averaging difference cycles. Obviously, each person will have a different manifold based on his spatio-temporal characteristics. Third, non-rigid transformation, using an approach similar to [25], is performed to find a unified manifold representation as in Fig. 5.2-b . Correspondences between different subjects are accomplished by selecting a certain body pose as the origin point in different manifolds and equal sampling in the parameterized representation. Finally, we parameterized the unified mean manifold by spline fitting.

The unified mean-manifold can be parameterized by a one-dimensional parameter $\beta_t \in R$ and a spline fitting function $f : \mathbb{R} \rightarrow \mathbb{R}^3$ that satisfies $b_t = f(\beta_t)$ which is used to map from the parameter space into the three dimensional embedding space. $b_t \in \mathbb{R}^3$ denotes the embedded coordinate of body configuration at time t. Such parameterization, along with the style parameterization enables generation of contours at different phases of the walking cycles and at different shape styles. Fig. 5.2-c shows three sequences generated using the same equidistant body configuration parameter $[\beta_1, \beta_2, \cdots, \beta_{16}]$ along the unified mean manifold with different style.

Embedding Kinematic Manifold

An alternative approach to reach a representation of the motion manifold invariant from visual variability is to use motion captured data to obtain an embedding of the kinematics manifold. We obtain a low dimensional representation of the kinematic manifold for gait by applying nonlinear dimensionality reduction techniques for motion-captured data. We first convert joint angles of motion-captured data into joint locations in a three-dimensional human-centered coordinate system. We aligned global transformation in advance in order to count motion only due to body configuration change. In order to find a low dimensional intrinsic representation from

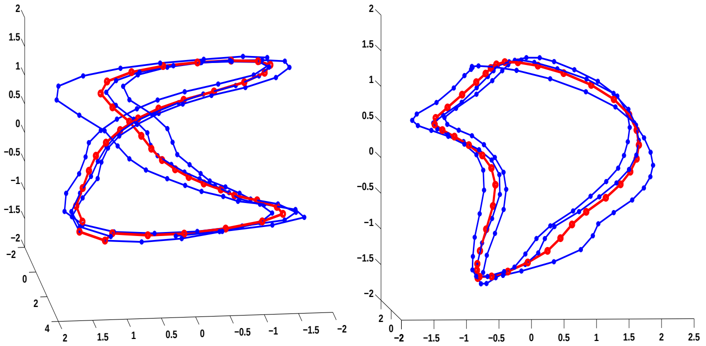


Figure 5.3: Kinematics manifold embedding and its mean manifold: two different views in 3D space

the high dimensional data (collection of joint location) we applied nonlinear dimensionality reduction procedure like Locally linear embedding (LLE) [111]. In order to find intrinsic manifold representation using nonlinear dimensionality reduction, dense sampling from the manifold points is required. Therefore, we used multiple cycles to find kinematics intrinsic manifold representation by LLE. Fig. 5.3 shows the kinematic manifold embedding based on three walking cycles of motion-captured data and their mean manifold representation. As with the case of visual data, the manifold is one-dimensional twisted closed manifold in three-dimensional spaces.

From multiple cycles, mean-manifold is computed and is parameterized by fitting a spline with an one-dimensional parameter $\beta_t \in \mathbb{R}$. A spline fitting function $f : \mathbb{R} \rightarrow \mathbb{R}^3$ that satisfies $\boldsymbol{b}_t = f(\beta_t)$ is used to map from the parameter space into the three dimensional embedding space as shown in Fig.5.3. Given such parameterization, any sequence of visual data (silhouettes), can be aligned to the kinematic manifold by finding cycles from the visual data and uniformly sampling the parameterized kinematic manifold according to the number of frames in each cycle.

5.3.2 Modeling Shape Style Space

Given a unified manifold embedding (whether obtained using visual or kinematic data), individual variations of the shape deformation can be discovered in the nonlinear mapping space between the embedding and the visual observation (silhouettes) for different people.

We learn a set of person dependent nonlinear mappings between the unified embedding space and shape sequences using Generalized Radial Basis Function (GRBF) [103]. The mapping has the form $\boldsymbol{y}_t^k = \gamma_k(\boldsymbol{b}_t) = \boldsymbol{C}^k \cdot \psi(\boldsymbol{b}_t)$, where $\boldsymbol{y}_t^k \in \mathbb{R}^D$ is person k shape at time t, and $\boldsymbol{b}_t$ is the corresponding point on the embedded manifold. $\boldsymbol{C}^k$ is a $D \times N$ mapping coefficients matrix which depends on particular person's shape. The nonlinear function $\psi(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^N$ defines an empirical kernel map [113] defined using N RBF kernel functions fitted to model the manifold in the embedding space (See details in Sec. 4.3. Here we gives brief summary.).

Given the learned nonlinear mapping coefficients $\boldsymbol{C}^1, \boldsymbol{C}^2, \cdots, \boldsymbol{C}^K$, for training people $1, \cdots, K$, the shape style parameters are decomposed by fitting an asymmetric bilinear model [128] to the coefficient space such that

$$[\boldsymbol{c}^1 \cdots \boldsymbol{c}^K] = \boldsymbol{A}\boldsymbol{S}, \tag{5.4}$$

where each $\boldsymbol{c}^k$ is a DN-dimensional vector representation of the matrix $\boldsymbol{C}^k$ using column stacking. The matrix $\boldsymbol{A}$ is an $DN \times K$ matrix containing the style basis for the coefficient space. The style matrix, $\boldsymbol{S} = [\boldsymbol{s}^1 \boldsymbol{s}^2 \cdots \boldsymbol{s}^k]^\mathsf{T}$, is an orthonormal matrix containing style vectors. Such decomposition can be obtained using Singular Value Decomposition. Therefore, the k-th person coefficient matrix $\boldsymbol{C}^k$ can be obtained from style vector $\boldsymbol{s}^k$ by restacking the vector $\boldsymbol{c}^k = \boldsymbol{A}\boldsymbol{s}^k$.

As a result, we can generate contour instance $\boldsymbol{y}_t^k$ for particular person k at any body configuration $\boldsymbol{b}_t$ using

$$\boldsymbol{y}_t^k = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{s}^k \times_2 \psi(\boldsymbol{b}_t), \tag{5.5}$$

where $\boldsymbol{\mathcal{A}}$ is a $D \times K \times N$ third order tensor (obtained by restacking the matrix $\boldsymbol{A}$). $\times_1$ and $\times_2$ are mode 1 and 2 tensor products as defined in [72]

Ultimately the style parameter $\boldsymbol{s}$ should be independent of the configuration and therefore should be time invariant and can be estimated at initialization. However, we don't know the person style initially and, therefore, the style needs to fit to the correct person style gradually during the tracking. So, we formulated style as a time variant factor that should stabilize after some frames from initialization. This will be described in Sec. 5.4.3.

Shape style vector $\boldsymbol{s}$ is a linear combination of the orthonormal basis of the style space. The dimension of the style vector depends on the number of people × cycles used for training and, therefore, can be high. The tracking of the high dimensional style vector s_t itself will be hard as it can fit local minima easily. Therefore, we represent any new style as a convex linear combination of style classes learned from the training data. A new style vector s is represented by linear weighting of each of the style classes s^q, $q = 1, \cdots, Q$ using linear weight $\lambda = [\lambda_1 \cdots \lambda_Q]$:

$$s = \sum_{q=1}^{Q} \lambda^q s^q, \qquad \sum_{q=1}^{Q} \lambda^q = 1, \tag{5.6}$$

where Q is the number of style classes used to represent new styles which are obtained by clustering the training data in the style space.

The overall generative model can be expressed as

$$\boldsymbol{z}_t = T_{\boldsymbol{\alpha}_t}\left(\mathcal{A} \times \left[\sum_{q=1}^{Q} \lambda_t^q \boldsymbol{s}^q\right] \times \psi(f(\boldsymbol{\beta}_t))\right). \tag{5.7}$$

We parameterize the geometric transformation $\boldsymbol{\alpha}$ by four parameters, scaling S_x, S_y and translation T_x, T_y. The tracking problem using this generative model is the estimation of parameter $\boldsymbol{\alpha}_t$, $\boldsymbol{\beta}_t$, and λ_t at each new frame given the observation $\boldsymbol{z}_t$.

5.4 Shape Style Adaptive Bayesian Tracking in Factorized Models

Given the shape generative model introduced above, the tracking problem is an inference problem where at time t we need to infer the body configuration $\boldsymbol{b}_t$ and the shape style $\boldsymbol{s}_t$ and the geometric transformation $T_{\boldsymbol{\alpha}_t}$ given the observation $\boldsymbol{z}_t$. The Bayesian tracking framework enables a recursive update of the posterior $P(\boldsymbol{x}_t|\boldsymbol{z}^t)$ over the object state $\boldsymbol{x}_t$ given all observations $\boldsymbol{Z}^t = \boldsymbol{z}_1, \boldsymbol{z}_2, .., \boldsymbol{z}_t$ up to time t:

$$P(\boldsymbol{x}_t|\boldsymbol{Z}^t) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{Z}^{t-1}) d\boldsymbol{x}_{t-1} \tag{5.8}$$

Where state $\boldsymbol{x}_t$ is the three sub-states, $[\boldsymbol{\alpha}_t, \boldsymbol{b}_t, \boldsymbol{s}_t]$, which uniquely describes the state of the tracking object. We represent three dimensional body configuration parameters $\boldsymbol{b}_t$ as a one-dimensional parameter β_t as explained in Sec. 5.3.1. The shape style is also parameterized by style class weighting parameters $\boldsymbol{\lambda}_t$ as in Sec. 5.4.3. For global transformation, we estimate geometric transformation parameters $\boldsymbol{\alpha}_t$ in the image space. So, using the generative model in Eq. 5.7, the tracking problem is to estimate $\boldsymbol{\alpha}_t$, $\boldsymbol{\lambda}_t$, and β_t for given observations $\boldsymbol{z}^t$.

5.4.1 Modeling Dynamics

We can model state dynamics by modeling the dynamics of each sub-states: dynamics of body configuration, dynamics of style state, and dynamics of global transformation. With the Bayesian framework in Eq. 5.8, and the graphical model in Fig. 5.1 the dynamic model $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ can be represented by

$$P(\boldsymbol{\alpha}_t, \boldsymbol{b}_t, \boldsymbol{s}_t|\boldsymbol{\alpha}_{t-1}, \boldsymbol{b}_{t-1}, \boldsymbol{s}_{t-1}) = P(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})P(\boldsymbol{b}_t|\boldsymbol{b}_{t-1})P(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}). \tag{5.9}$$

For body configuration, since we parameterize embedding configuration manifold by a one-dimensional spline parameter β_t as in Sec. 5.3, the parameter β_t will change in a constant speed if the subject walks in a constant speed (because it corresponds to a constant frame rate used in the learning). However, the resulting manifold point representing body configuration $\boldsymbol{b}_t = f(\beta_t)$ will move along the manifold at different step sizes. That is a fundamental reason why we use β_t, rather than $\boldsymbol{b}_t$, to model the dynamics since it results in a one-dimensional constant-speed linear dynamic system. In general, the walking speed can change gradually. So, the body configuration in each new state will move from the current state with a filtered speed, $\tilde{v}$, that can be adaptively estimated during tracking. Therefore, the new frame body configuration β_t is modeled as one dimensional Gaussian around $\beta_{t-1} + \tilde{v}$ with variance σ_b^2.

The dynamic model of the style sub-state is approximated by a random walk since the style may change smoothly around a specific person style. Given the parameterization of style space introduced in Sec. 5.3.2, the style at time t is modeled as multivariate Gaussian around λ_{t-1} with covariance Σ_s, i.e.,

$$P(s_t|s_{t-1}) \propto P(\lambda_t|\lambda_{t-1}) = N(\lambda_{t-1}, \Sigma_s)$$

The global transformation α_t captures global contour motion in the image plane (translation and scaling).

5.4.2 Particle Filtering

The objective is to estimate the state posterior given observations. The decomposable feature of our generative model enables us to estimate each substate by a marginal posterior distribution, i.e., we keep three state posteriors $P(\boldsymbol{\alpha}_t|\boldsymbol{z}^t)$, $P(\boldsymbol{b}_t|\boldsymbol{z}^t)$, and $P(\boldsymbol{s}_t|\boldsymbol{z}^t)$.

We approximate the marginal density of each sub-state using maximum a posteriori (MAP) of the other sub-states, i.e.,

$$P(\boldsymbol{\alpha}_t|\boldsymbol{z}^t) \propto P(\boldsymbol{\alpha}_t|\boldsymbol{b}_t{}^*, \boldsymbol{s}_t{}^*, \boldsymbol{z}^t), \;\; P(\boldsymbol{b}_t|\boldsymbol{z}^t) \propto P(\boldsymbol{b}_t|\boldsymbol{\alpha}_t{}^*, \boldsymbol{s}_t{}^*, \boldsymbol{z}^t), \;\; P(\boldsymbol{s}_t|\boldsymbol{z}^t) \propto P(\boldsymbol{s}_t|\boldsymbol{\alpha}_t{}^*, \boldsymbol{b}_t{}^*, \boldsymbol{z}^t),$$

where $\boldsymbol{\alpha}_t{}^*$, $\boldsymbol{b}_t{}^*$, and $\boldsymbol{s}_t{}^*$ are MAP estimate of each approximated marginal density.

We represent state densities using particle filters since such densities can be non-Gaussian and the observation is nonlinear. Given the parameterization of the sub-states in terms of $\boldsymbol{\alpha}_t$,β_t,λ_t, the marginalized posterior densities are approximated by three particle systems.

$$\{\boldsymbol{\alpha}_t^{(i)}, {}^{\alpha}\pi_t^{(i)}\}_{i=1}^{N_\alpha}, \{\beta_t^{(j)}, {}^{\beta}\pi_t^{(j)}\}_{j=1}^{N_b}, \{\boldsymbol{\lambda}_t^{(k)}, {}^{\lambda}\pi_t^{(k)}\}_{k=1}^{N_s}, \tag{5.10}$$

where N_α,N_b, and N_s are the numbers of particles used for each sub-states and ${}^{\alpha}\pi_t^{(i)}$,${}^{\beta}\pi_t^{(i)}$,${}^{\lambda}\pi_t^{(i)}$ are their corresponding weights.

5.4.3 Style Estimation with Constraints and Annealing Procedure

There are two factors to be considered in shape style estimation: First the high dimensionality of the style representation. The parameterization of the style space in terms of convex linear weights of a small number of style classes, as in Eq. 5.6, reduces the dimensionality of the style space.

Second, the style estimation needs to become more discriminative as tracking progresses. At the beginning, we don't know the correct style. To avoid being trapped in local minima, we start from the mean style, which is the style with uniform weights for all the representative shape style classes. As additional evidence (frames) becomes available, the estimated style vector can gradually be more discriminative so that weighting particles become more sensitive to observations.

To achieve this progressive discrimination, we use a deterministic annealing like procedure: estimated style weights are forced to be close to uniform weights at the beginning to avoid hard decisions about style classes and gradually become discriminative thereafter using a temperature parameter. We controlled the variance of style from large to small value to control discrimination in the weighting estimation of styles.

To achieve this, the re-weighting of the style particle is controlled by a temperature parameter that controls how the weights are influenced by the observation. We assume the observation distribution given the style particle s, global transformation $\boldsymbol{\alpha}_t^*$ and body configuration $\boldsymbol{b}_t^*$, can be approximated by a Gaussian distribution.

$$^{\lambda}\pi_t^{(k)} \propto P(\boldsymbol{z}_t|\boldsymbol{\alpha}_t^*, \boldsymbol{b}_t^*, \boldsymbol{s}_t^{(k)}) \propto \exp\left(-\frac{d(\boldsymbol{z}_t, \boldsymbol{z}_t^{(k)})^2}{\Sigma_t^2}\right) = \exp\left(-\frac{d(\boldsymbol{z}_t, T_{\boldsymbol{\alpha}_t^*}\boldsymbol{\mathcal{A}} \times \boldsymbol{s}_t^{(k)} \times \psi(\boldsymbol{b}_t^*))^2}{\Sigma_t^2}\right),$$

where $d(\cdot)$ is distance measure as described in Sec. 5.4.5, $\boldsymbol{z}_t^{(k)}$ is the contour from the generative model using $\boldsymbol{\alpha}_t^*, \boldsymbol{b}_t^*, \boldsymbol{s}_t^{(k)}$. When the variance Σ_t^2 is very big ($\Sigma_t^2 >> d(\boldsymbol{z}_t, \boldsymbol{z}_t^{(k)})^2$), the weight $^{s}\pi_t^{(k)}$ will be assigned similar value regardless to $d(\cdot)$. When the variance is small ($\Sigma^2 < d(\boldsymbol{z}_t, \boldsymbol{z}_t^{(k)})^2$), the likelihood is sensitive to the distance value and corresponding weights in the particle update will be discriminative. To achieve annealing-like procedure, we use style class variances, which are uniform to all classes and are defined by $\Sigma^s = T_s\sigma_s^2 + \sigma_o$ respectively as time variant parameters. The parameters T_s start with large values at the first frame and are gradually reduced and in each step and a new body configuration estimate is computed.

5.4.4 Tracking Algorithm

We perform tracking by sequential update of the marginalized sub-densities utilizing the predicted densities of the other sub-states. These densities are updated with current observation $\boldsymbol{z}_t$ by updating weighting values of each sub-state particle approximations given observations. We estimate global transformation $\boldsymbol{\alpha}_t$ using predicted estimates $\hat{\boldsymbol{s}}_t^*, \hat{\boldsymbol{b}}_t^*$. Then body configuration $\boldsymbol{b}_t$ is estimated using estimate global transformation $\boldsymbol{\alpha}_t^*$, and predicted style estimate $\hat{\boldsymbol{s}}_t^*$. Finally style $\boldsymbol{s}_t$ is estimated with given the estimates for $\boldsymbol{\alpha}_t^*$, and $\boldsymbol{b}_t^*$. The following table summarizes the state estimation procedure using time $t-1$ estimation.

1. Importance-sampling with re-sampling at $t-1$:

For given $t-1$ state density estimation: $\{\alpha_{t-1}^{(i)}, {}^{\alpha}\pi_{t-1}^{(i)}\}_{i=1}^{N_\alpha}, \{\beta_{t-1}^{(j)}, {}^{b}\pi_{t-1}^{(j)}\}_{j=1}^{N_b}, \{\lambda_{t-1}^{(k)}, {}^{s}\pi_{t-1}^{(k)}\}_{k=1}^{N_s}$.

Re-sampling: $\{\grave{\alpha}_{t-1}^{(i)}, 1/N_\alpha\}$, $\{\grave{\beta}_{t-1}^{(j)}, 1/N_b\}$, and $\{\grave{\lambda}_{t-1}^{(k)}, 1/N_s\}$.

2. Predict current state densities using dynamic models:

$\alpha_t^{(i)} = H\grave{\alpha}_{t-1}^{(i)} + N(0, \sigma_\alpha^2)$

$\beta_t^{(j)} = \grave{\beta}_{t-1}^{(j)} + \tilde{v}_t + N(0, \sigma_b{}^2), \qquad b_t^{(j)} = f(\beta_t^{(j)})$

$\lambda_t^{(k)} = \grave{\lambda}_{t-1}^{(k)} + N(0, \sigma_{s}{}_{t-1}^2), \qquad \lambda_t^{(k)} = \frac{\lambda_t^{(k)}}{\sum_{i=1}^{N_s} \lambda_i^{(k)}{}_t},$

3. Force style particle to satisfy constraints of Eq. 5.6:

If $\lambda_i^{(k)} \leq 0$ then, $\lambda_i^{(k)} = 0$ for all i,k , $\qquad \lambda_t^{(k)} = \frac{\lambda_t^{(k)}}{\sum_{i=1}^{N_s} \lambda_i^{(k)}{}_t},.$

4. Sequential update of state weights using current observation:

Global transformation α_t with $\hat{b}_t$, $\hat{s}_t$:

$P(\alpha_t^{(i)}|\hat{b}_t^*, \hat{s}_t^*, z_t) \propto P(z_t|\alpha_t^{(i)}, \hat{b}_t^*, \hat{s}_t^*)P(\alpha_t^{(i)})$

${}^{\alpha}\pi_t^{(i)} = P(z_t|\alpha_t^{(i)}, \hat{b}_t^*, \hat{s}_t^*), \quad {}^{\alpha}\pi_t^{(i)} = \frac{{}^{\alpha}\pi_t^{(i)}}{\sum_{j=1}^{N_\alpha} {}^{\alpha}\pi_t^{(j)}}$

Body pose b_t with α_t, $\hat{s}_t$:

$\alpha_t^* = \alpha_t^{(i^*)}$, where $i^* = \arg\max_i {}^{\alpha}\pi_t^{(i)}$

$P(b_t^{(j)}|\alpha_t^*, \hat{s}_t^*, z_t) \propto P(z_t|\alpha_t^*, b_t^{(j)}, \hat{s}_t^*)P(b_t^{(j)})$ ${}^{b}\pi_t^{(j)} = P(z_t|\alpha_t^*, b_t^{(j)}, \hat{s}_t^*),$ ${}^{b}\pi_t^{(j)} = \frac{{}^{b}\pi_t^{(j)}}{\sum_{i=1}^{N_b} {}^{b}\pi_t^{(i)}}$

Style s_t with α_t, b_t:

$b_t^* = b_t^{(j^*)}$, where $j^* = \arg\max_j {}^{b}\pi_t^{(j)}$

$P(s_t^{(k)}|\alpha_t^*, b_t^*, zt) \propto P(zt|\alpha_t^*, b_t^*, s_t^{(k)})P(s_t^{(k)})$

${}^{s}\pi_t^{(k)} = P(z_t|\alpha_t^*, b_t^*, s_t^{(k)}), \quad {}^{s}\pi_t^{(k)} = \frac{{}^{s}\pi_t^{(k)}}{\sum_{i=1}^{N_s} {}^{s}\pi_t^{(i)}}$

5. Reducing style variance:

5.4.5 Observation Models

In our multi-state representation, we update weights ${}^{\alpha}\pi_t^{(i)}$, ${}^{b}\pi_t^{(j)}$, and ${}^{s}\pi_t^{(k)}$ by marginalized likelihood $P(z_t|\alpha_t^{(i)}, b_t^*, s_t^*)$, $P(z_t|\alpha_t^*, b_t^{(j)}, s_t^*)$, and $P(z_t|\alpha_t^*, b_t^*, s_t^{(k)})$ which can be measures given observation z_t. Each sub state captures different characteristics of the dynamic motion and affects different variations in the observation. For example, body contour shape changes according to body configuration state. Different body configurations show significant changes in edge direction at the legs. However, for case of style, the variation is subtle and changes along the global contours. Therefore, we use three different observation models for each of the marginalized likelihoods above based on suitable distance measure for each component. We use three distance measures: *Chamfer distance*, *weighted Chamfer distance*, and *oriented Chamfer distance*.

Representation

We represent the generated contours by an implicit functions where the contour is the zero level of such function. From each new frame z_t, we extracted edge using Canny edge detector and distance field is computed which is used to compare implicit shape representation of model generated contours. Fig. 5.4 shows an example of edge detection and distance transformation for detected edge.

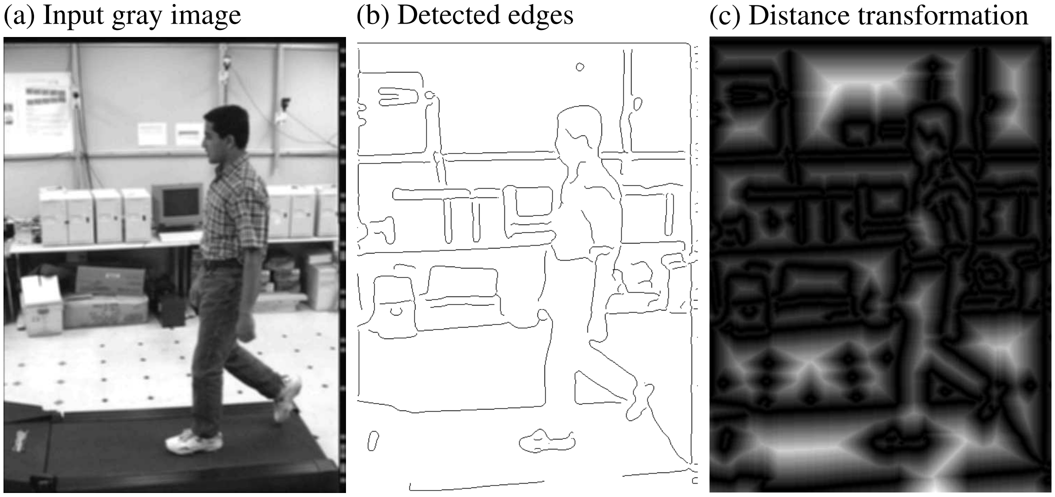


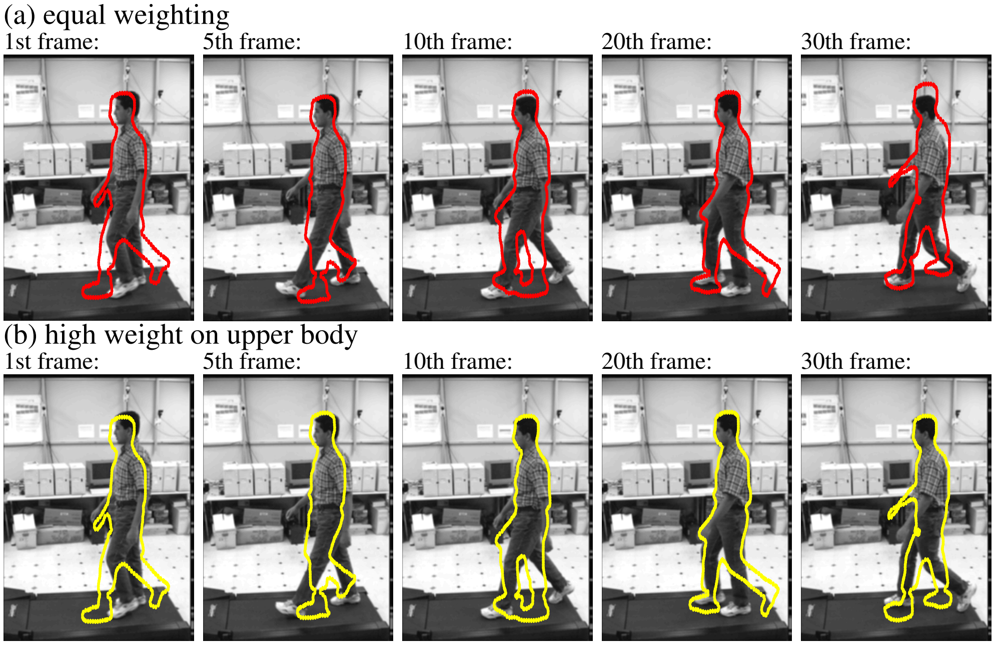
Figure 5.4: An example of edge detection and corresponding distance transformation for an input image

Weighted Chamfer distance for geometric transformation estimation

For geometric transformation estimation, the predicted body configuration, and style estimate from the previous frame are used. Therefore, we need to find similarity measurement which is robust to the deviation of body pose and style estimation and sensitive to global transformation. Typically the shape or the silhouette of upper body part in walking sequence are relatively invariant to the body pose and style. By giving different weight to different contour points in Chamfer matching, we can emphasize upper body part and de-emphasize lower body part in the distance measurement. Weighted chamfer distance can be computed as

$$d_w(T,F,W) = \frac{1}{N}\sum_{i}^{N} \min_{f_i \in F} \rho(t_i, f_i) w_i, \tag{5.11}$$

where t_i is i'th feature location, f_i template feature and w_i i'th feature weight. Practically, weighted chamfer distance achieved more robust estimation of the geometric transformation. Fig. 5.5 shows efficiency of weighted chamfer matching even inaccurate body pose estimation. We applied known style and added offset to known body configuration parameter in global transformation estimation. In case of equal weighting, it failed to accurate tracking. However, it shows robust tracking even in accurate body pose when we weighted only upper body part in Chamfer matching.



(a) equal weighting

(b) high weight on upper body

Figure 5.5: Geometric transformation estimation :Inaccurate shape style value and body configuration value with offset from true value are used in order to evaluate robustness on inaccurate body configuration estimation

Oriented Chamfer distance for body pose

Different body poses through walking can be characterized by the orientation of legs. Therefore, oriented edge based similarity measurement is useful in case of the body configuration estimation. Oriented chamfer distance matching was used in [41]. We use a linear filter [44] to detect oriented edge efficiently after edge detection. After applying the linear filter to the contour and the observation, we applied chamfer matching for each oriented distance transform and oriented contour template. The final result is the sum of each of the oriented chamfer distance. We used multiplication of chamfer distance measurement and oriented chamfer distance as distance measure in the body configuration estimation. For style estimation, simple Chamfer distance is used. Fig. 5.6 shows four different oriented edge detection and corresponding distance transformation result.

5.5 Experimental Results

We evaluated the performance of proposed style adaptive gait tracking algorithm using CMU Mobo gait data set, M. Black's walking sequence, and Southampton gait database.

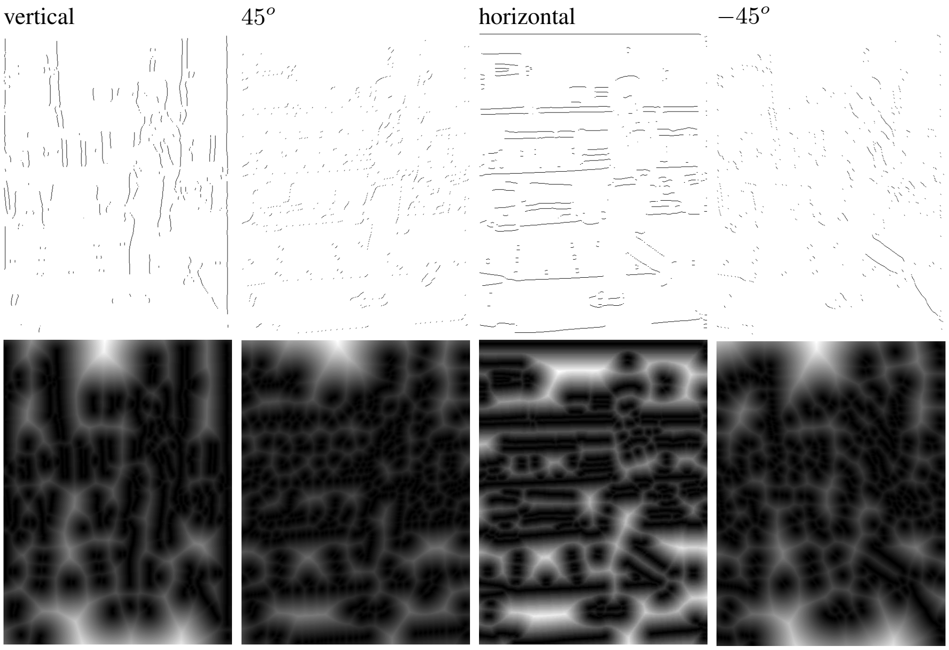


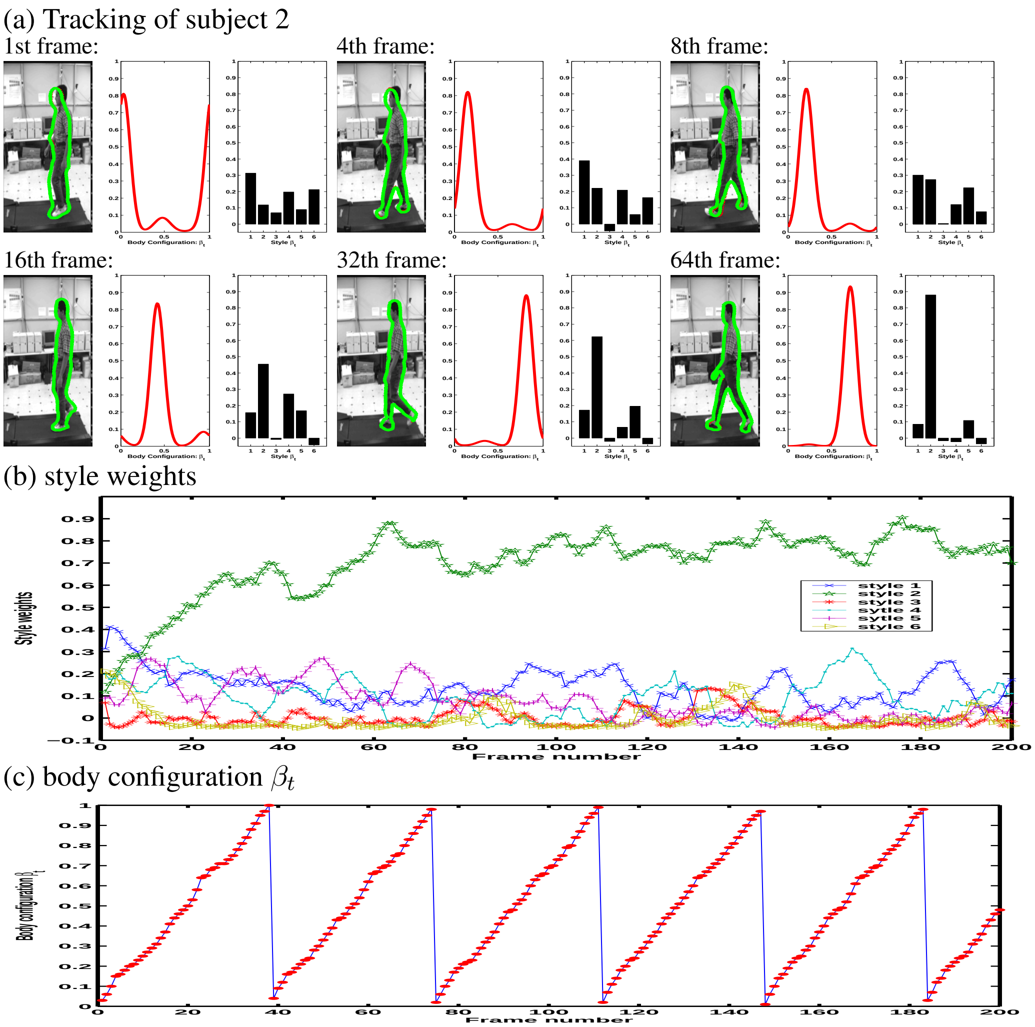
Figure 5.6: Oriented edge detection and corresponding distance transformation

5.5.1 Gait Tracking Using Visual Manifold Embedding

We used CMU Mobo data set for learning the generative model, the dynamics, and for testing the tracker. Six subjects are used in learning the model. The tracking performance is evaluated for people used in training and for unknown people, which were not used in the learning. We initialize the tracker by giving a rough estimate of initial global transformation parameter. The body configuration is initialized by random particles along the manifold. In case of style, as we don't know the subject style from the initial frame, the tracker is initialized by mean style, which means equal weights are applied for every style class.

Tracking for trained subjects

For subjects in the training data, i.e. the shape style has been seen before in training, the tracker shows very good tracking results. It shows accurate tracking of body configuration parameter β_t and correct estimation of shape style s_t. Fig. 5.7 (a) shows several frames during tracking known subject. In each sub-figure the left column shows tracking contours. The red color contour shows predicted contour from previous frame after geometric transformation and the green color shows after updating the style using current observation and estimated body



(a) Tracking of subject 2

1st frame: 4th frame: 8th frame:

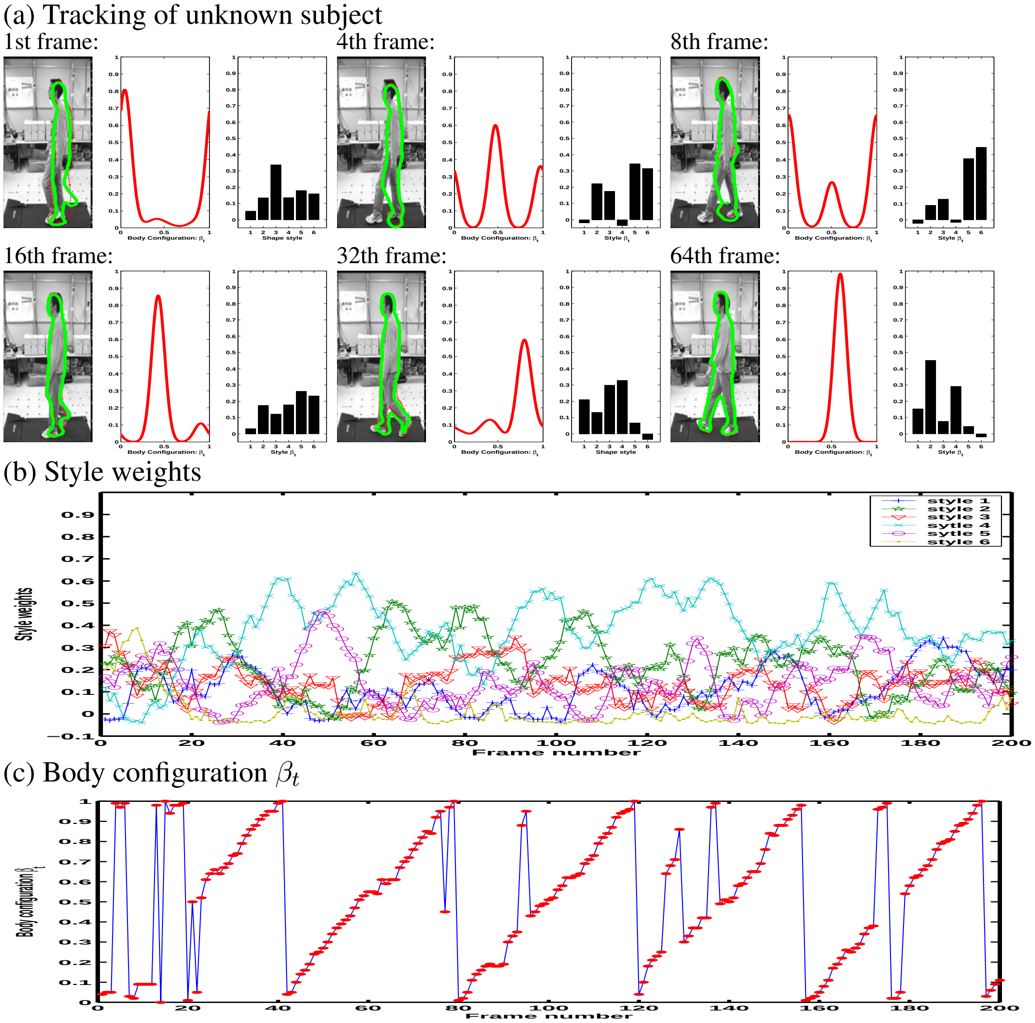
16th frame: 32th frame: 64th frame:

(b) style weights

(c) body configuration β_t

Figure 5.7: Tracking for known person

configuration. The middle column shows the posterior body configuration density. The right column is the estimated style weights in each frame. Fig. 5.7 (b) shows tracking results for style weights. The figure shows that the style estimate converges to the subject's correct style and it becomes the major weighting factor after about 10 frames. The style weighting shows accurate identification of the subject as a result of tracking and it has many potential for human identification and others. Fig. 5.7 (c) shows estimated body configuration β value. Even though the two strides making each gait cycle are very similar and hard to differentiate in visual space, the body configuration parameter accurately finds out the correct configuration. The figure shows that the body configuration correctly exhibits constant speed linear dynamics. As we have one to one mapping between body configuration on the manifold and 3D body pose, we can directly recover 3D body configuration using the estimated β or using manifold points b_t similar to [36].



(a) Tracking of unknown subject

(b) Style weights

(c) Body configuration β_t

Figure 5.8: Tracking for unknown person

Tracking for unknown subjects

Tracking for new subjects can be hard as we used small number of people for learning style parameters in the generative model. Such subjects' shape styles will be linear combination of trained subjects' styles It takes more frames to converge to accurate contour fitting as shown in Fig. 5.8 (b). However, after some frames it accurately fit to the subject contour even though we did not do any local deformation to fit the model to the subject. There is no one dominant style class weight during the tracking and sometimes the highest weight are switched depend on the observation. In case of the body configuration β_t, you can see sometimes it jumps about half cycle due to the similarity in the observation since the style is not accurate enough. Still, the result shows accurate estimation of body pose.

We also tested the tracker in normal walking situation using an out door sequence [1] based on the model trained from CMU Mobo gait data. Even though our system learned the generative model from treadmill walking data, proposed system can perform accurate contour tracking in normal walking. Fig. 5.9 shows contour tracking results for 40 frames. Fig. 5.9 (c) shows estimated body configuration parameters. It confused in some intervals at the beginning but it recovers within the cycle.

5.5.2 Gait Tracking and Recognition Using Kinematic Manifold Embedding

We evaluated the performance of the proposed algorithms on University of Southampton (UoS) gait database [118]. The database provides well-extracted silhouette images under controlled environments for walking sequence of more than 100 people. We used provided silhouette sequences to learn our nonlinear generative model. We collected 10 subjects to learn the global shape deformations dependent on individual style and embeddings. Four cycles from each person are used to learn the style variations in each person. Total 40 cycles are used to learn the generative model ($N_s = 40$) after kinematics manifold embedding.

Table 5.1: Gait recognition confusion matrix:():percentage (%)

Person Id	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	**1(3.3)**	0	25 (83.3)	0	3(10)	1(3.3)	0	0	0	0
P2	0	**30(100)**	0	0	0	0	0	0	0	0
P3	0	0	**30(100)**	0	0	0	0	0	0	0
P4	0	0	0	**23(76.7)**	0	3(10)	0	2(6.7)	0	2(6.7)
P5	0	0	0	1(3.3)	**28(93.3)**	0	0	1(3.3)	0	0
P6	1(3.3)	1(3.3)	14(46.7)	0	0	**0**	3(10)	0	0	0
P7	0	0	0	0	0	0	**24(80.0)**	5(16.7)	0	1(3.3)
P8	0	0	1(3.3)	0	0	0	1(3.3)	**28(93.3)**	0	0
P9	1(3.3)	0	25(83.3)	0	0	0	1(3.3)	2(6.7)	**0**	1(3.3)
P10	0	0	11(36.7)	0	0	0	1(3.3)	3(10)	0	**15(50)**

Synthesis of new dynamic shapes

We tested the performance of synthesis of shape deformation according to shape style vector in our nonlinear generative model by changing style parameter and its dimension. Fig. 5.10 (a)

[1] http://www.cs.brown.edu/people/black/

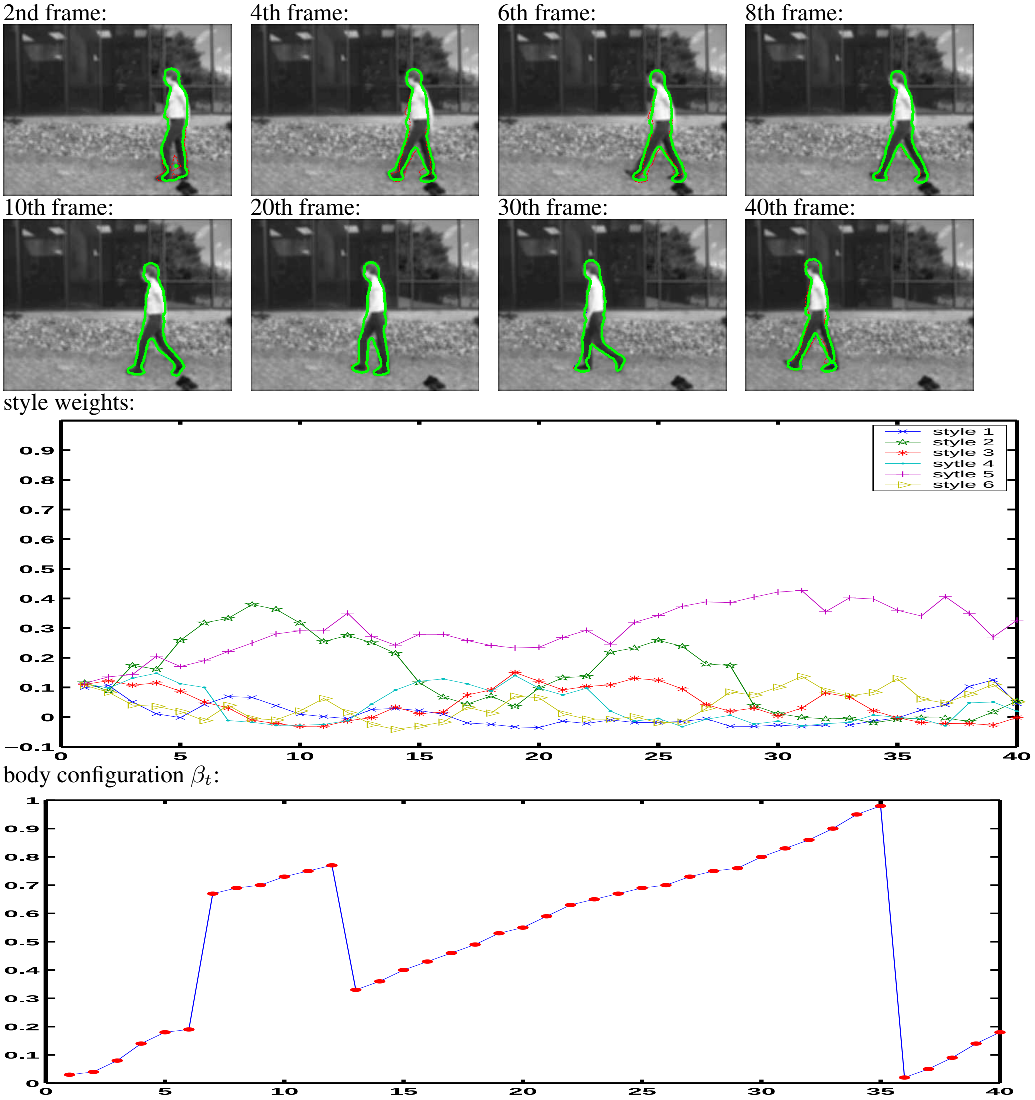
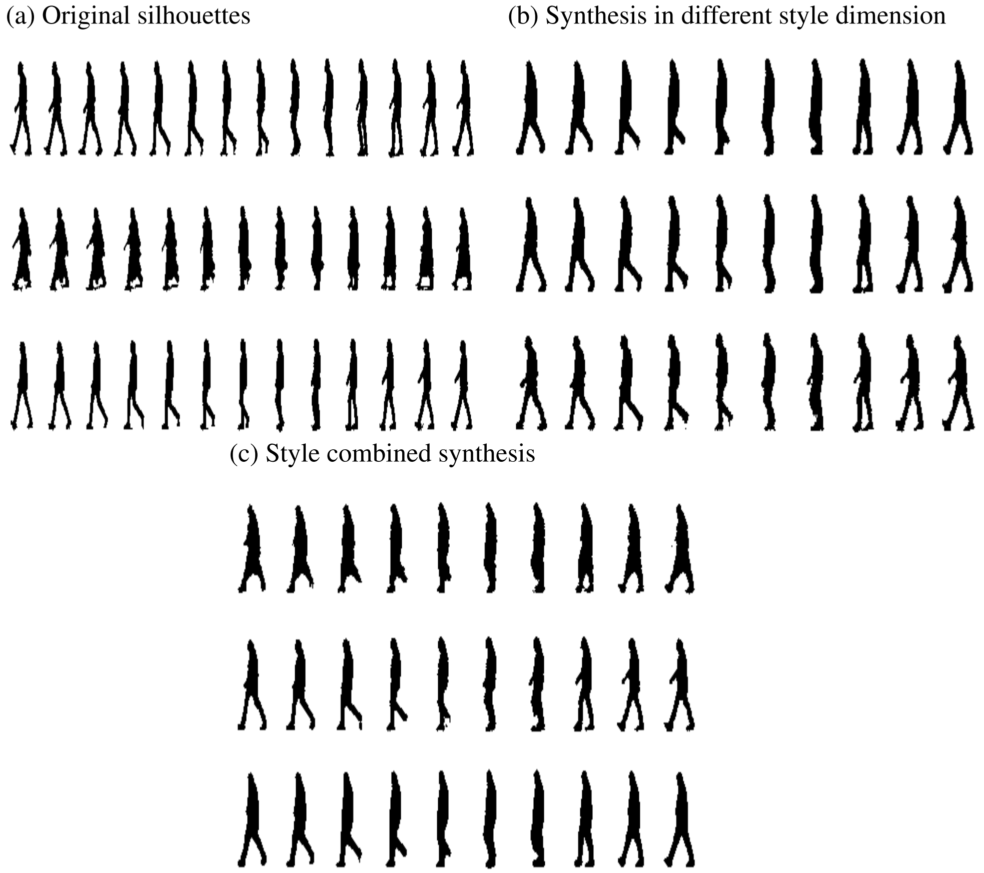


Figure 5.9: Tracking straight walking

shows collected original sequence of three different people. When we use reduced number of style basis, we lost details of the person. However, we are still able to generate sequences showing body pose changes even with one basis as shown in the first row of Fig. 5.10 (b). When we used corresponding person style vectors with full dimension, the new sequence preserves detail difference of individual shape deformation. Fig. 5.10 (c) shows linear interpolation of style vector and corresponding shape interpolation. This capability allows tracking of new people adaptively as shown in the following experiments. In addition, although the original silhouette sequences have different number of frame in each cycle, we can control the synthesized shapes to be aligned as we used the same manifold points to synthesize body configurations.



(a) Original silhouettes

(b) Synthesis in different style dimension

(c) Style combined synthesis

Figure 5.10: Style dependent dynamic shape synthesis: (a) Row 1: P1, Row 2: P2, Row 3: P3 original silhouette, (b) Synthesis of P1 silhouettes using Row 1: 1 style basis, Row 2: 25% style basis, Row 3: full style basis, (c) Synthesis by style combination: Row 1: 0.5P1+0.5P2, Row 2: 0.5P1+0.5P3, Row 3: combinations of all style vectors equally (mean style vector)

Recognition using shape style

We tested the performance of gait recognition in two situations. First, we perform gait recognition during tracking using edge information without any background subtraction. Gait recognition is performed by selecting the style with highest weighted particle. We tested the gait recognition performance for indoor sequences. The indoor sequences have relatively simple background. However, when we use just edge information, it is not easy to estimate the whole shape and identify the person as it has many missed edge in the corresponding contours and additional edge inside desired contour, which causes confusion in the estimation of shape using edge-based distance transformation (DT). Fig. 5.11 shows change of style weights for two people gait sequences. In both of the case, the weights begin from equal weights and gradually fit to one of the shapes. In case of person 2, the person style get dominant quickly. In the case

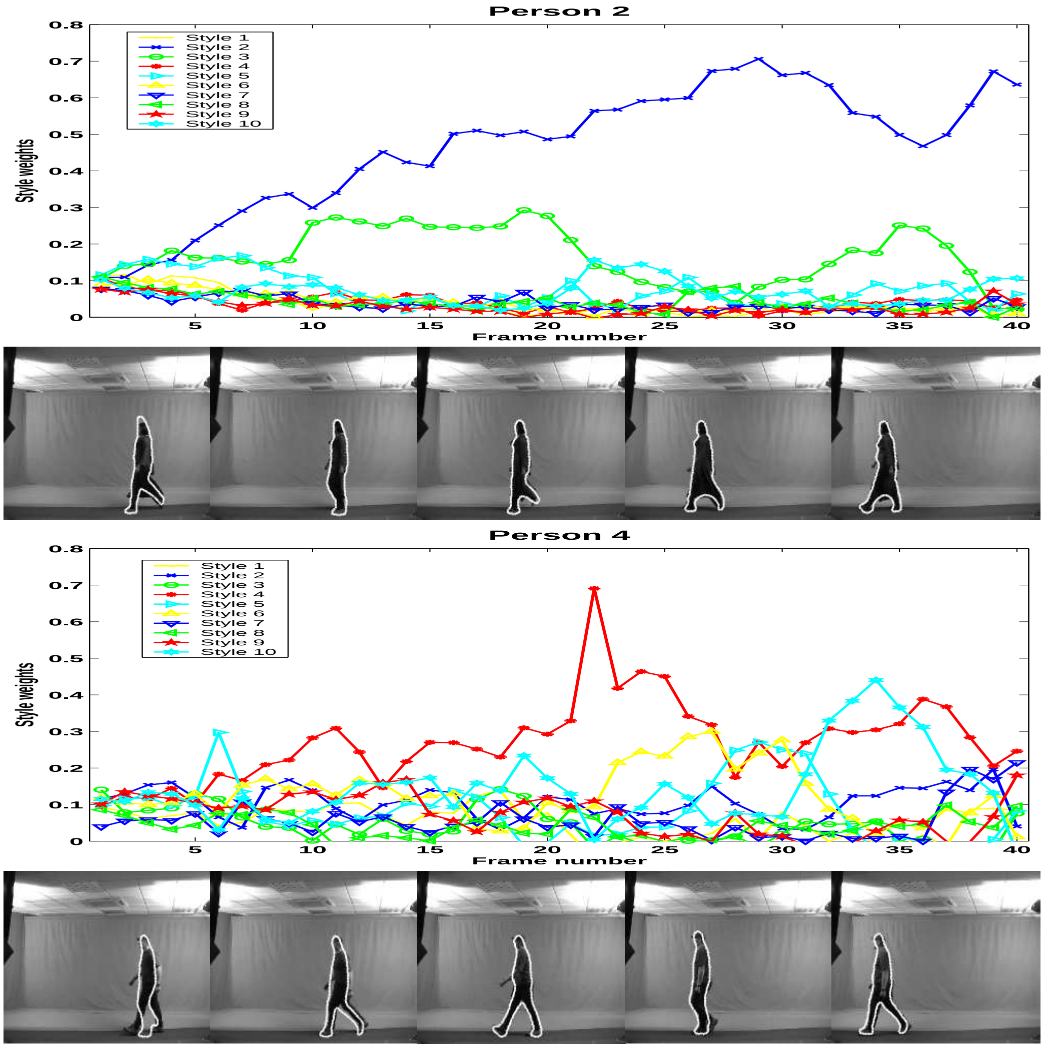


Figure 5.11: Simultaneous gait recognition and tracking:Left: person 2 style weights and contour tracking at 5th, 10th, 20th, 30h, and 40th frames. Right: person 4 style weights and contour tracking at 5, 10, 20, 30, 40th frame

of person 4, the style estimate fails to find correct style when the geometric transformation misaligned contours around 30^{th} frame. But, overall for most of the sequence, the tracker adapted to the subject shape correctly. Table 5.1 shows gait recognition results from each person sequence. We did not count style weights of the initial 10 frames as style weights are not reliable at the beginning.

Second, we tested the gait recognition when extracted silhouette sequences are given. We selected 4 cycles from 37 people and learn the generative model. In this case, the style dimension becomes 148 (37×4) dimension. We collected another 3 cycles which are not used for model learning from the same database and estimated style vector in closed form using pseudo

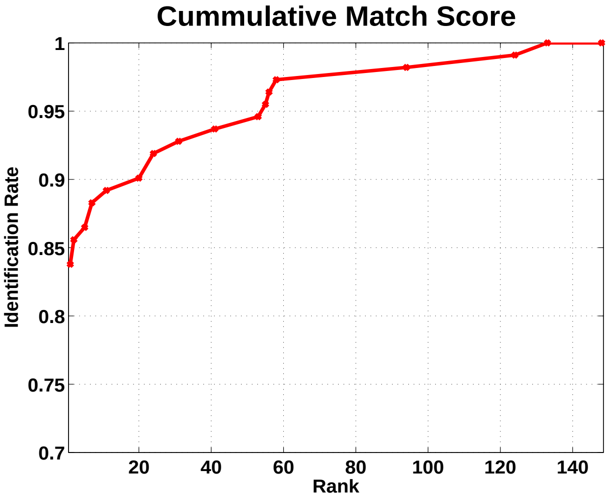


Figure 5.12: Performance of gait recognition using style vector

inverse. For each estimated 148 dimensional vector, we compute similarity by inner product $\boldsymbol{S} \cdot \boldsymbol{s}^{est}$, which gives cosine value of two vector since the style basis are orthonomal. We classified gait by maximum similarity value and we get 83.8% recognition rate from 37 subjects by recognizing 93 sequence correctly at rank 1 among 111 (3×37) sequences. Further experiment shows cumulative matching characteristics(CMC) as in Fig. 5.12.

5.6 Summary

We presented new framework for human motion tracking and recognition using decomposable generative models. Using manifold embedding and parameterization, we can perform tracking body pose on a one-dimensional manifold. By representing variations in spatiotemporal contour deformation among different people using style vectors, we can achieve person identification using gait simultaneously with person-adaptive contour tracking. For accurate estimation of high dimensional style vector, we added constraints in the shape style particles and employed annealing-like gradual increase of discrimination. As a result of our tracking, we not only find accurate contour from cluttered environment without background learning or appearance model, but also get parameters for body configuration and shape style.

In this chapter, we assumed fixed view in learning the generative model and tracking human motion. We need to extend the model to continuous view variant situations. We used marginalized density approximation instead of full joint distribution of the state. Sampling based on Markov Chain Monte Carlo (MCMC) can be used for more accurate estimation of the marginal density. We performed gait recognition with simple similarity measurement and relatively small dataset that showed promising results. More advanced classification algorithms can be performed using style vectors as feature vectors.

Chapter 6

Modeling View and Posture Manifold

We model shape deformations corresponding to both view point and body configuration changes through the motion. Such observed shapes present a product space (different configurations × different views) and lie on a low dimensional manifold in the visual input space. The approach we introduce here is based on learning both the visual observation manifold and the kinematic manifold of the motion in a supervised manner. We learn the geometric deformation between an ideal manifold (conceptual equivalent topological structure) and a twisted version of the manifold (the data). We use a torus manifold to represent such data for both periodic and non-periodic motions. Experimental results show accurate estimation of 3D body pose and view from a single camera using the torus manifold. In addition, We propose modeling a manifold within the mapping space of another manifold. We utilize both the observation manifold and the 3D kinematics manifold to learn a generative model with two independent continuous manifold parameterizations, one for the body configuration and one for the view variations. The resulting representation is used for tracking complex motions within a Bayesian framework where the model provides a low dimensional state representation as well as a constrained dynamic model for both body configuration and view variations.

6.1 Overview: Learning Continuous View Manifold

Despite the high dimensionality of the body configuration space, many human motion activities lie intrinsically on low dimensional manifolds. Exploiting such property is essential to constrain the solution space for many problems such as tracking, posture estimation, and activity recognition. For many motions, such as gait, kicking, golf swing, gestures, etc., the body configuration changes along a one dimensional manifold, which can be closed for periodic motion as walking or running, or it can be open trajectory in motions such as golf swing or kicking.

It follows that, the observed motion, in terms of body shape contour and/or appearance lie on a low dimensional manifold as well. However, the observed motion manifold changes given the view point.

We consider tracking and inferring view and body configuration of human motion from a single monocular camera where the person can change his/her view with respect to the camera while bing tracked (or equivalently the camera can be moving). Modeling both the view and body configuration manifolds for human motion jointly in the visual space is a very challenging task and is useful for tracking, posture estimation, and view estimation. On the other hand, we assume a simple setting for view variations. We consider the motion being observed from different view points along a view circle at a fixed camera height, i.e, we restrict the view manifold to be one dimensional.

The approach we introduce here is based on learning the visual observation manifold in a supervised manner. Traditional manifold learning approaches are unsupervised where the goal is to find a low dimensional embedding of the data. However, if the manifold topology is known the manifold learning can be formulated in a different way. Manifold learning is then the task of learning a mapping from/to a topological structure to/from the data where that topological structure is homeomorphic to the data. In this paper we argue that this supervised setting is suitable to model human motions that lie intrinsically on a one dimensional manifolds whether closed and periodic such as walking, jogging, running, etc., or open such as golf swing, kicking, tennis serve, etc. We show that we can model the visual manifold of such motions (in terms of shape) as observed from different view points by mapping such manifold to a torus manifold. We also consider this problem for general motions, i.e., we do not restrict ourselves to one dimensional motion manifold as in [92]. So, fundamentally, the approach can handle complex motions.

We propose a framework for modeling both the configuration and view manifolds. We use kinematics manifold as a representation of the configuration invariant to view. Given an embedding of the kinematic manifold, the view manifold is then explicitly modeled in the nonlinear mapping space between the kinematics manifold embedding and the view-variant observations. The result is two low-dimensional embeddings: one for configuration and one for the view, as well as a generative model that can generate observation given the two manifolds' parameterizations. This fits perfectly into the Bayesian tracking as it provides in a direct way: 1) low dimensional state representation for both view and configuration, 2) a constrained dynamic model since the manifolds are modeled explicitly, 3) an observation model, which comes directly from the generative model used.

6.2 Framework

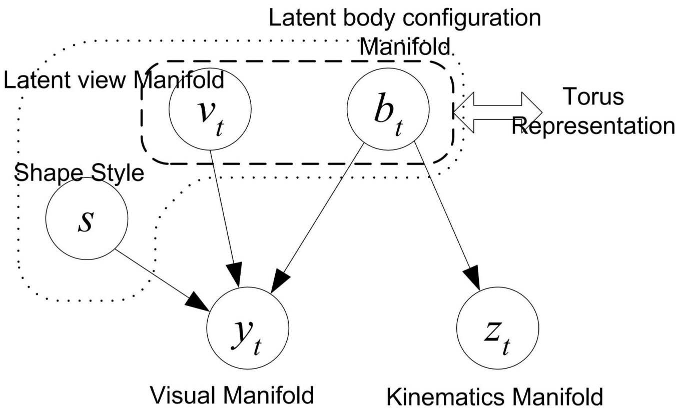


Figure 6.1: Graphical Model

Consider a motion observed from a camera (stationary or moving). Such motion can be represented as a kinematic sequence $\boldsymbol{Z}^T = \boldsymbol{z}_1, \cdots, \boldsymbol{z}_T$ and observed as a sequence of observation $\boldsymbol{Y}^T = \boldsymbol{y}_1, \cdots, \boldsymbol{y}_T$. In this paper, by observation, we mainly mean shape contours. With an accurate 3D body model, camera calibration, and geometric transformation information, we can explain $\boldsymbol{Y}^T$ as a projection of an articulated model. The dynamic sequence $\boldsymbol{Z}^T$ lies on a manifold, let's call it kinematic manifold. Also, the observations lie on a manifold, visual manifold. In fact, observations are lying on a product manifolds, the body configuration and the view manifolds.

What is the relation between the kinematic manifold and the visual input manifold. We can think of a graphical model connecting the two manifolds through two latent variables: body configuration variable, $\boldsymbol{b}_t$ and a view point variable, $\boldsymbol{v}_t$. The body configuration variable is shared between both the kinematic manifold and the visual manifold. The view point variable represents the relative camera location to a human centered coordinate system. Another variable affecting the observation is the shape variability among different subjects.We denote this variable by s, which is time invariant variable.

We can summarize our goals as follows:

1) We need to relate the kinematic manifold with the visual input manifold in order to be able to infer configuration from input

2) We need model the visual manifold with all its variabilities due to the motion, the view point, and shape style. In particular, we need to be able to deal with both body configuration and view points as a continuous variables. This facilitates tracking subjects with varying view points due to camera motion or changing subject view w.r.t. the camera.

3) We need the tracking state space to be low dimensional and continuous. Moreover, and despite the nonlinearity in dynamics in both the kinematics and the observations, we need the model to exhibits simple dynamics, i.e., linear dynamics or even constant speed dynamics

So, let us start with a simple periodic motion such as a simple aerobic exercise or gait, observed from a view circle around the person. Later we show how to deal with more complex motions and also extend to the whole view sphere. Given a set of observed shapes representing a product space of two one-dimensional manifolds representing body configuration and view, how can we learn a useful representation. Nonlinear manifold learning techniques, such as LLE [111], Isomap [127], etc., have been popular recently in learning low dimensional representations of both visual and kinematic data. Unfortunately such techniques are limited when dealing with complex manifolds such as joint motion and view manifolds and will not necessarily lead to useful representations. This can be observed in Fig. 6.2 (d),(e) where LLE and Isomap are used to embed data with continuous view and configuration variability as shown in Fig. 6.2 (a). The resulting embedding, although reflects the actual manifold local structure, is not useful as a representation for tracking. Moreover, if we consider different people, the joint manifold is expected to twist differently depending on the shape of the person performing the motion. Therefore, the resulting representation will not be useful to generalize to other people. The conclusion is the data-driven embedding of the joint view-configuration manifold is not practical to be used in tracking, synthesis, or analysis tasks.

Supervised Generative Manifold Learning:

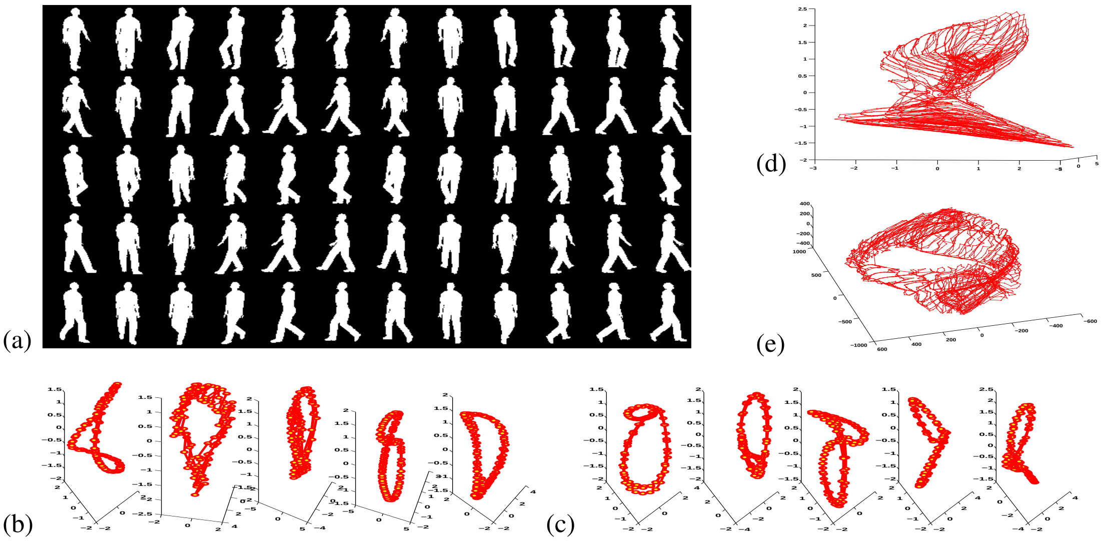


Figure 6.2: Data-driven view and body configuration manifolds:(a) Examples of sample data with view and configuration. Rows: body pose at $0, \frac{1}{5}T, \frac{2}{5}T, \frac{3}{5}T, \frac{4}{5}T$. Cols:view $0, 30, 60, \cdots, 330$. (b) Intrinsic configuration manifold when view angle is $0, 60, 120, 180, \text{and} 240$. (c) View manifold for five different fixed body pose. (d) (e) Combined view and body configuration manifold by LLE and Isomap.

Traditional manifold learning approaches are unsupervised where the goal is to find a low dimensional embedding of the data which preserve the manifold topological structure. However, if the manifold topology is known, manifold learning can be formulated in a different way. Manifold learning is then the task of learning the deformation of the manifold from an ideal case. For example, for the gait case, observed from the same view point, as shown in the examples in Fig. 6.2 (b), the gait manifold is a one dimensional closed manifold which is topologically equivalent to a unit circle. So, we can think of the gait manifold as a twisted or deformed circle in the visual input space. Since we already know the topology, the task of manifold learning can be viewed as: how to deform a unit circle to reach the actual data manifold. Or, in other words, how to generate the data knowing an equivalent "idealistic" topological structure. In fact, this view can be even extended if the data manifold does not share the exact topology from the ideal manifold. For example, the gait manifold can intersect itself in the visual space but still, we can learn the deformation from a unit circle to the data. Similarly, if we consider the view manifold for a certain body posture, the resulting manifolds are topologically equivalent to unit circle as can be seen in Fig. 6.2 (c).

For the case of joint configuration and view manifold where the view varies along a view circle, this is a product space and ideally is equivalent to the produce of two circles, i.e., torus manifold. i.e., the data in Fig. 6.2 (a) lies on a deformed torus in the input space. So we need to learn deformation from the torus to the data. If we consider the full view sphere, the resulting manifold is a deformed order-3 torus or $S1 \times S1 \times S1$ structure.

On the other hand, the kinematic manifold, which is invariant to view point, is also a deformed circle in the kinematic space. Starting from a torus, the kinematic manifold can be achieved through collapsing the torus along one of its axis to form a circle and then deform that circle. Therefore, a torus manifold acts as an "ideal" manifold to represent both the latent body configuration and view variables, $\boldsymbol{b}_t$, $\boldsymbol{v}_t$. In one side, the torus can deform to form the visual manifold, $\boldsymbol{y}_t$, and on the other side, it can deform to form the kinematic manifold $\boldsymbol{z}_t$.

Configuration and View Manifold:

In a complex motion, such as aerobics or dance routines, the body configuration cannot be represented in one dimensional manifold as in the torus manifold. In order to solve this problem, we consider two separate manifolds: 1) the body configuration manifold during the motion in the kinematics space 2) the visual input manifold (the observations) of the same motion observed from different view points along a view circle at a fixed camera height. It is clear that the kinematics manifold can be embedded using nonlinear dimensionality reduction techniques to achieve a low dimension representation of the manifold which can be used for tracking. For example, Gaussian Process Dynamic Models (GPDM) [141] can achieve such embedding as well as learn dynamic model for such manifold. The challenge is the visual manifold since it involves both body configuration and view variability. Embedding such complex manifold will not result in useful representation and definitely will not facilitate inference about the configuration and view separately.

In fact, giving camera setting, the observation for a given body posture lies on a one dimensional manifold (view manifold) in the visual input space. Obviously, each body posture will have it's own view manifold. If we consider a sequence of postures, making up a motion, the resulting visual manifold well become complicated as it becomes a product of the motion manifold and the view manifold.

6.3 Torus Manifold: View and Configuration Joint Representation

6.3.1 Torus Manifold Embedding

Torus Manifold

A torus manifold, a two dimensional manifold embedded in three dimensional space with a single hole, is useful to represent both periodic and non-periodic dynamic human motion observed from a viewing circle.

The torus manifold can be constructed from a rectangle, which can be represented by two orthogonal coordinates with range $[0 \quad 1] \times [0 \quad 1]$, by gluing both pairs of opposite edges together with no twists [46]. Therefore, the torus surface can be parameterized by two variables $u, v \in [0 \;\; 1]$.

As justified in Sec. 6.2 the torus can be used as a conceptual embedding for the joint view (along one viewing circle) and configuration manifold. The view and body configuration manifold can be parameterized in the rectangle coordinate with the two orthogonal axis of the torus manifold. Any manifold point in the torus can have two circles: one is in the plane of the torus, which we use to model the view variable and parameterized with μ, and the other is perpendicular to it which we use to represent the body configuration and parameterized by ν.

Generalization to the full view sphere around the person is straight forward. In this case the joint configuration and view manifold can be mapped to a family of tori, which is a subset of the product space of three circles $S1 \times S1 \times S1$, one for the body configuration, one for the horizontal view circle and one for the vertical view circle. In practice, only small range of the vertical view circle is considered, therefore, this can be modeled as a set of rectangles each representing a torus manifold for a given view circle, i.e., can be parameterized by three parameters μ, ν, ξ for body configuration, view angle and elevation view angle.

How to embed points on the torus

Given a sequence of kinematic data $\boldsymbol{Z}$ representing a motion, we can use graphics software to render body silhouettes from different view points along a given viewing circle. We denote this data by $\boldsymbol{Y}$. It is desired to embed this data on the torus in a conceptual way that does not necessarily reflect their Euclidean distance in the kinematic space nor in the visual input space, instead the objective is to embed them on the torus in a way to simplify the tracking. There are two ways we can achieve such embedding.

Constant Speed Dynamics: For tracking, we not only know the topology of the manifold but we may also know the desired dynamics in the state space. For example, for periodic motion such as walking and running, although the nonlinearity in dynamics in both the kinematic and the visual input manifolds, we need the latent state variable to reflect a constant speed on the latent manifold. The nonlinear mapping in Eq. 6.1 should transform this linear dynamics to nonlinear dynamics. This can be achieved by embedding the points on equidistance points along the configuration axis of the torus.

Geodesics-based Embedding: For non-periodic motion, such as golf swing, where data might exhibit different acceleration along the course of the motion, it is desired to embed the data on the torus in a way that preserves their kinematic manifold structure. This can be achieved through embedding the points such that the geodesic distance on the torus is proportional to the geodesics on the kinematic manifold. Another constraint stems from the fact that in non-periodic motion, the manifold is an open trajectory and therefore, configuration manifold should be mapped to a part of the torus configuration axis.

To achieve this, we first embed the kinematic manifold using LLE or any other nonlinear embedding techniques. This leads to an open trajectory embedding. Such embedding is used for 1) measuring the gab between the beginning and end pose of the motion in order to map the manifold to a proportional part of the torus. 2) to measure the geodesics along the kinematic manifold. The points are embedded on the torus in such a way that only a part of the torus ν axis is used proportional to the embedded manifold length. Let $\boldsymbol{x}_i, i = 0, \cdots, N$ be the embedding coordinate of the kinematic sequence $\boldsymbol{z}_i, i = 0, \cdots, N$. The coordinate of point $\boldsymbol{z}_i$ on the torus ν-axis is denoted by ν_{z_i} and is set to be $\nu_{\boldsymbol{z}_i} = S_i/S$ where S_i is the geodesic distance of point x_i to the starting point, x_o, i.e., $S_i = \sum_{j=1}^{N} ||\boldsymbol{x}_j - \boldsymbol{x}_{j-1}||$ and S is defined to be $S = S_N + ||\boldsymbol{x}_N - \boldsymbol{x}_o||$. The gap between the beginning body pose embedding point and final body pose embedding points on the torus will be $Gap = \frac{||\boldsymbol{x}_N - \boldsymbol{x}_0||}{S}$. Fig. 6.3 (a) shows an example a golf swing motion from three different view points and it low dimensional embedding of the kinematics using LLE is shown in (b). Fig. 6.3 (c) shows a torus manifold with a gap between the start and the end body pose embedding for the case of a golf swing.

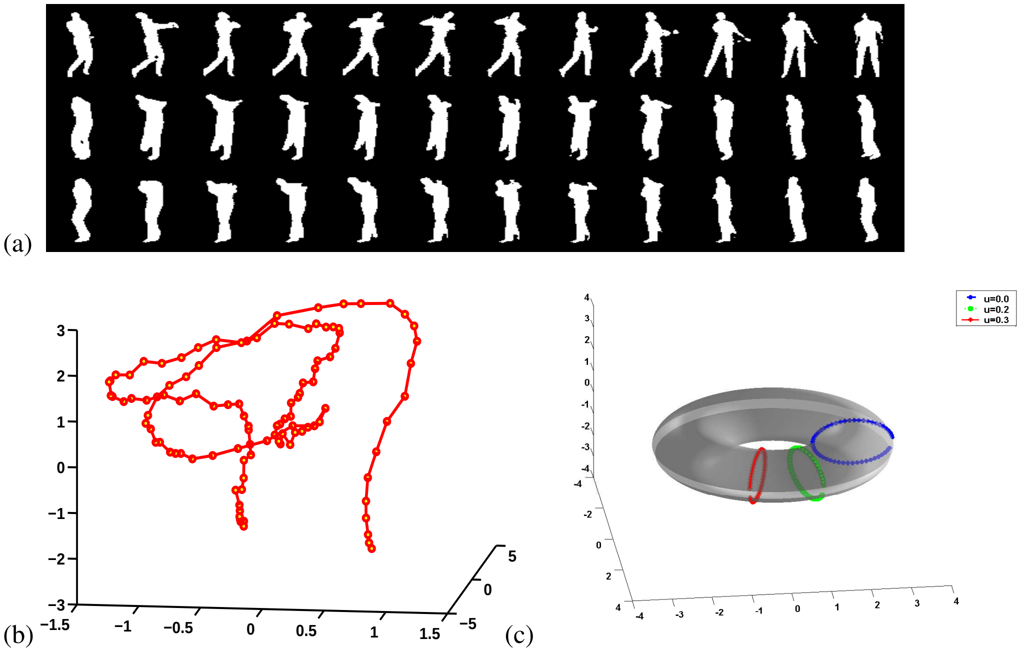


Figure 6.3: Torus manifold with gap. (a) Example sequence of a golf swing from three different views $\mu = 0, 0.2, 0.3$. (b) Embedding of golf swing motion capture data. (c) Visualization of a torus manifold with gap with trajectories of the three different views used for synthesis in (a)

6.3.2 Learning Manifold Deformation

Learning a mapping from a topological structure to the data, where that topological structure is homeomorphic to the data, can be achieved through learning a regularized nonlinear warping function. Let $\mathcal{T}$ denotes the torus manifold and $\mathcal{M}$ denotes a data manifold where $\mathcal{T}$ and $\mathcal{M}$ share the same topology. Given a set of point $\boldsymbol{x}_i \in \mathbb{R}^d, i = 1 \cdots, K$ on $\mathcal{T}$ and their corresponding points $\boldsymbol{y}_i \in \mathbb{R}^D, i = 1 \cdots, K$ on a manifold $\mathcal{M}$, we can learn a nonlinear mapping function $g : \mathbb{R}^d \rightarrow \mathbb{R}^D$ from $\mathcal{T}$ to $\mathcal{M}$. According to the representer theorem [71], such function admits a representation in the form of $\boldsymbol{y} = \sum_j b_i k(\boldsymbol{x}, \boldsymbol{z}_j)$ where $\boldsymbol{z}_j$ are a finite set of points in the input space, not necessarily data points, and $k(.,.)$ is a kernel function. If radial basis kernels are used then this is a form of radial basis function interpolation. Given the data embedded on the torus as described above, we can learn the deformation between the torus and both the visual manifold and the kinematic manifold. This can be achieved through learning two regularized nonlinear mapping functions in the form of Eq. 6.1 as follows:

Torus to Visual Manifold: Deforming the torus to the visual manifold can be achieved through learning a nonlinear mapping in the form of Eq. 6.1. Given the embedding coordinates on the torus, (μ_v, ν_b) and their corresponding visual data (silhouettes), $\boldsymbol{y}^{vb} \in \mathbb{R}^D$, for discrete pose $\boldsymbol{b}$ and view $\boldsymbol{v}$, we can fit the mapping function $g : \mathbb{R}^2 \rightarrow \mathbb{R}^D$ which map from the torus to the shape space in the form

$$\boldsymbol{y} = g(\mu, \nu) = \boldsymbol{D} \cdot \psi(\mu, \nu) \tag{6.1}$$

satisfying $\boldsymbol{y}^{vb} = g(\mu_v, \nu_b)$. In this case, we need a set N of basis functions covering the torus surface which are set uniformly across the surface. Using this model, for any view $\boldsymbol{v}$ and body configuration $\boldsymbol{b}$ sequence, we can generate a new observations where μ_v is view representation in μ axis, and ν_b is body configuration representation in ν axis of the torus manifold.

Torus to Kinematic Manifold:

Deforming the torus to the kinematic manifold can be achieved through learning a nonlinear mapping from the torus configuration axis to the kinematic manifold. Given the embedding on the torus, (μ_v, ν_b) and their corresponding kinematic points $\boldsymbol{z}_b \in \mathbb{R}^d$ we fit the mapping function $f : \mathbb{R} \rightarrow \mathbb{R}^d$ in the form

$$\boldsymbol{z} = f(\nu) = \mathbf{B} \cdot \psi(\nu) \tag{6.2}$$

stratifying that $\boldsymbol{z}^b = f(\nu_b)$. Given this mapping, any point on the torus (μ, ν) can be directly mapped to a 3D joint position configuration.

Learning Different People Manifolds from Partial Views

Our goal is to be able to achieve adaptive tracking where the tracker can adapt to the person contour shape. We presented an approach for decomposing “style” variations in the space of nonlinear mapping coefficients from an embedded manifold to the observation space in Chapter 4. Similar approach can be used here to learn style dependent mappings in the form of Eq. 6.1 from the torus to each person’s data. The torus represents a unified manifold representation invariant to the person.

Given different people sequences from different sparse view points, each sequence can be embedded on the torus as described in Sec. 6.3.1. Let $\boldsymbol{Y}^s_{\boldsymbol{v}_k}$ be sequences of visual data for person $\boldsymbol{s}$ from view points $\boldsymbol{v}_k$, we can embed such sequences on the torus which leads to a set of torus coordinates $(\mu_{\boldsymbol{v}_k}, \nu_b)$. The view points do not need to be the same across subjects and the sequences do not need to be the same length; only the beginning and end of the motion is needed to be aligned on the torus ν-axis. Given the embedding points and their corresponding contours, person-specific mapping functions in the form of Eq. 6.1 can be fitted which leads to an $D \times N$ coefficient matrix $\boldsymbol{D}^s$. Notice that the kernel space defined by $\psi(\cdot)$ in Eq. 6.1 is the same across all subjects since the same RBF basis are used on the torus. Given the coefficient matrices, we can fit a model in the form of

$$\boldsymbol{y}_{vb}^s = \mathcal{A} \times_1 \boldsymbol{a}^s \times_2 \psi(\mu_v, \nu_b) \tag{6.3}$$

where $\boldsymbol{a}^s$ is a vector characterizing the person shape style and $\mathcal{A}$ is third order tensor with dimensions $D \times S \times N$ where S is the dimensionality of the shape space and $\times_n$ is the tensor multiplication as defined in [72]. The model proposed here, provides a continuous representation of the view point and the body configuration in one latent representation space.

6.4 Posture-Invariant View Manifold Using Kinematic Manifold

6.4.1 Learning Configuration and View Manifold

Learning View-invariant Configuration Manifold

As a common representation of body configuration invariant to view point, we use an embedding of the kinematic manifold, which represents body configuration in a low dimensional space. Such kinematic manifold embedding is also invariant to different people shapes and appearances. We can obtain a low dimensional representation of the kinematic manifold by applying nonlinear dimensionality reduction for motion capture data using approaches such as LLE [111], Isomap [127], GPLVM [75], etc. In particular, we used LLE in this paper.Since we need to achieve embedding of the kinematics invariant to the person transformation with respect to the world coordinate system, we represent the kinematics using joints' location in a human-centered coordinate system. We aligned for global transformation in advance in order to only count motion due to body configuration changes.

Fig. 6.4(a) shows the embedded kinematic manifold for gait motion. As expected, for a periodic motion as in the gait case, the embedding shows the kinematic manifold as one dimensional twisted closed manifold which can be embedded free of intersections in a three dimensional embedding space. However, for more complex motion, the manifold is not necessarily be one dimensional. However, we can always achieve an embedding of the kinematic manifold in a low dimensional Euclidean space. Fig. 6.5 shows example embedding for the ballet dance routine data which is shown in Fig. 6.2.

Learning Posture-invariant View Manifold

Given the kinematic manifold embedding, we can achieve a representation of different views by analyzing the coefficient space of nonlinear mappings between the kinematic manifold embedding and view-dependent observation sequences.

Here, we can consider the kinematic manifold embedding as "content" manifold and the view can be thought as the "style" factor, where, such "style" variations are decomposed in the space of nonlinear mapping coefficients from an embedded manifold to the observation space. Unlike Chapter 4 where the content manifolds were view-dependent, the use of the kinematic manifold provides a view invariant content representation and therefore, differences between view-dependent observed data will be preserved in the nonlinear mapping of each view-dependent input sequences.

Given a set of N body configuration embedding coordinates on the kinematic manifold, $X = \{\boldsymbol{x}_1 \cdots \boldsymbol{x}_N\}$ and their corresponding view-dependent shape observations (silhouettes) $\boldsymbol{Y}^k = \{\boldsymbol{y}_1^k \cdots \boldsymbol{y}_N^k\}$ for each view k where $k = 1, \cdots, V$, we can fit view-dependent regularized nonlinear mapping functions on the form of generalized radial basis function

$$\boldsymbol{y}^k = \boldsymbol{B}^k \psi(\boldsymbol{x}), \tag{6.4}$$

for each view k. Here, each observation $\boldsymbol{y}$ is represented as $\boldsymbol{D}$ dimensional vector and we denote the embedding space dimensionality by e. $\psi(\cdot)$ is an empirical kernel map [113] $\psi_{N_c}(\boldsymbol{x}) : \mathbb{R}^e \rightarrow \mathbb{R}^{N_c}$ defined using N_c kernel functions centered around arbitrary points $\{\boldsymbol{z}_i \in \mathbb{R}^e, i = 1 \cdots N_c\}$ along the kinematic manifold embedding where

$$\psi_{N_c}(\boldsymbol{x}) = [\phi(\boldsymbol{x}, \boldsymbol{z}_1), \cdots, \phi(\boldsymbol{x}, \boldsymbol{z}_{N_c})]^{\mathsf{T}}. \tag{6.5}$$

Each $D \times N_c$ matrix $\boldsymbol{B}^k$ is a view-dependent coefficient matrix which encodes the view variability. Given such view-dependent mapping coefficients, we can fit a model in the form

$$\boldsymbol{y}_i^k = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{v}^k \times_2 \psi(\boldsymbol{x}_i), \tag{6.6}$$

where $\boldsymbol{\mathcal{A}}$ is a third-order tensor with dimensionality $D \times V \times N_c$ and $\times_j$ is the mode-j tensor multiplication [72]. This equation represents a generative model to synthesize observation vector $y_i^k \in \mathbb{R}^D$ of view k and configuration i given a view vector $\boldsymbol{v}^k$, and body configuration represented by embedding coordinate $\boldsymbol{x}_i \in \mathbb{R}^e$ on the kinematic manifold embedding.

To fit such model, the view-dependent coefficient matrices $\boldsymbol{B}^k, k = 1, \cdots, V$ are stacked as columns in a $(DN_c) \times V$ matrix $\boldsymbol{C}$ and then the view factors are decomposed by fitting an asymmetric bilinear model [128]. i.e., $\boldsymbol{C} = A \cdot [\boldsymbol{v}^1 \cdots \boldsymbol{v}^V]$. The third-order $(D \times V \times N_c)$ tensor $\boldsymbol{\mathcal{A}}$ in Eq. 6.6 is the tensor representation of the matrix A which can be obtained by unstacking its columns.

The resulting representation of the view variations is discrete and high dimensional. The dimensionality of the view vector in Eq. 6.6 depends on the number of views, i.e., V dimensional. This high dimensional representation is not desirable as a state representation for tracking. The dimensionality can be reduced when fitting the asymmetric model by finding fewer number of view bases. Fig. 6.4 (b) and Fig. 6.5 (b) show the embedded posture-invariant view manifold in the mapping coefficient space for gait and ballet dance motion respectively, which clearly shows a one dimension manifold that preserves the proximity between near by views. Here, the first three dimensions are shown. The actual view manifold can then be explicitly represented as will be shown in Sec. 6.4.2.

Learning Observation Variability

The model in Eq. 6.6 can be further generalized to include variable for shape style variability between different people, i.e., to model different people shapes. The use of the kinematic manifold provides an invariant representation to observation variabilities which allows us to generalize the model. Given view-dependent shape observations for different people, we can fit view-dependent, person-dependent mapping functions in the form of Eq. 6.4 which yields a set of coefficient matrices B^{kl} for each person l and view k. Given such coefficient matrices,

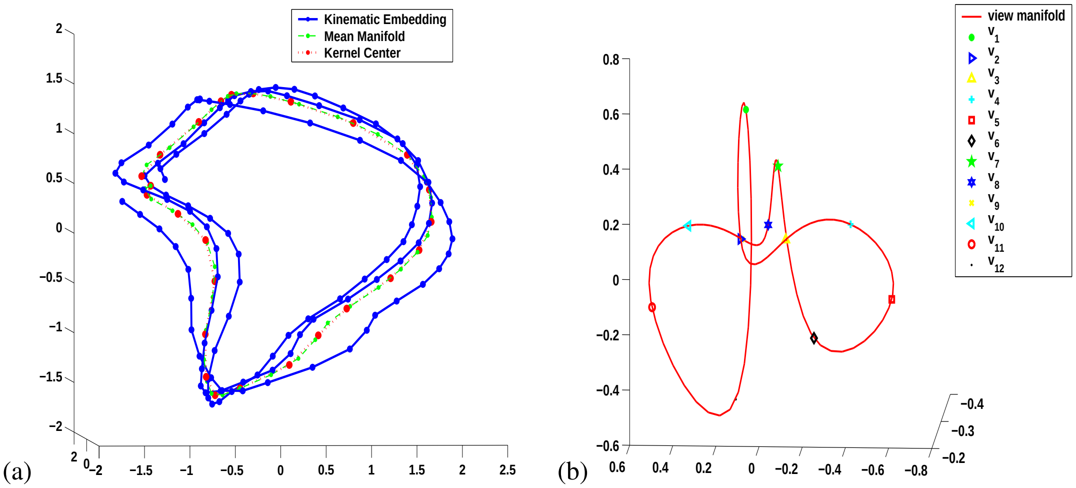


Figure 6.4: Configuration and view manifold for gait:(a) Embedded kinematics manifold. (b) Posture invariant view manifold (The first three dimensions are shown).

we can fit a generalized model in the form

$$\boldsymbol{y}_i^{kl} = \boldsymbol{\mathcal{D}} \times_1 \boldsymbol{s}^l \times_2 \boldsymbol{v}^k \times_3 \psi(\boldsymbol{x}_i), \tag{6.7}$$

where $\boldsymbol{\mathcal{D}}$ is a forth-order tensor with dimensionality $D \times S \times V \times N_c$. This equation represents a generative model to synthesize observation vector $y_i^{kl} \in \mathbb{R}^D$ of view k, shape style l and configuration i given a view vector $\boldsymbol{v}^k \in \mathbb{R}^V$, a shape style vector $\boldsymbol{v}^k \in \mathbb{R}^S$, and body configuration represented by embedding coordinate $\boldsymbol{x}_i \in \mathbb{R}^e$ on the kinematic manifold embedding. Fitting such model can be achieved using HOSVD [72, 138]

6.4.2 Parameterizations of View and Configuration Manifolds

Parameterizing the View Manifold

Given the view space defined by the decomposition in Eq. 6.6, different view vectors is expected to lie on a low dimensional nonlinear manifold. Obviously, linear combination of view vectors in Eq. 6.6 will not result in valid view vectors. We need to explicitly model the view manifold in the coefficient space to be able to predict and synthesize new views. Therefore, we model view variations as a one dimensional nonlinear manifold by one dimensional continuous variable using spline fitting with C^2 connectivity constraints between the last sample and the first sample view, since the view manifold is closed one dimensional manifold. As a result, we represent the view manifold by one dimensional view configuration parameter θ where certain view v_t

can be represented as $\boldsymbol{v}_t = g_v(\theta_t)$. Fig. 6.4 (b) and 6.5 (b) show spline parameterized one dimensional view manifold in three dimensional space.

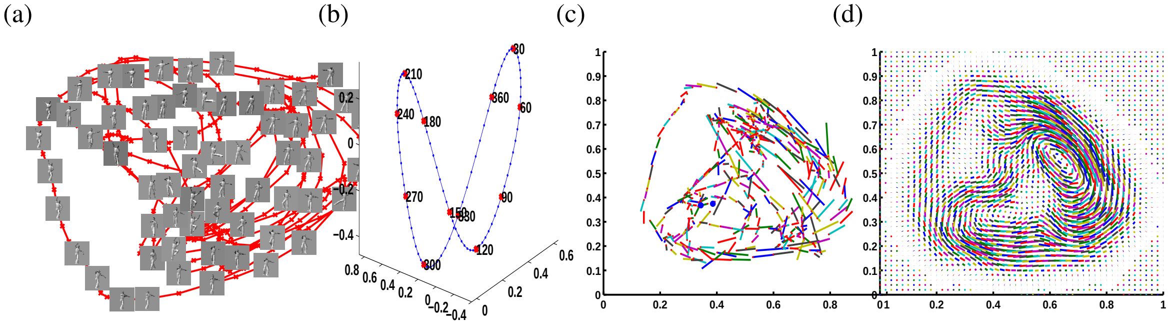


Figure 6.5: Configuration and view manifold for Ballet: (a) Embedded kinematics manifold in 2D for ballet sequence. (b) One dimensional view manifold embedded in the kinematic manifold mapping space (The first three dimensions are shown). (c) Velocity field on the manifold for ballet. (d) Interpolation of the velocity field value for any points on the manifold

Parameterizing the Configuration Manifold

In general, we make no assumption about the dimensionality of the motion manifold. However, here we discriminate between two cases: 1) the case of an one-dimensional motion, whether periodic, such as walking, running, etc., or non periodic open trajectory, such as golf swings, tennis serves, etc. 2) the case of a general motion where the actual motion manifold dimensionality is not know, e.g., dance or aerobics, etc.

Parameterizing One-Dimensional Motion Manifold

Since, in this case, the kinematic manifold is one dimensional, it can be represented using a one-dimensional parameter by spline fitting. we can represent view manifold and body configuration manifold using two continuous parameters θ_t and β_t and generate new observations jointly as:

$$\boldsymbol{y}_t^{vb} = \boldsymbol{\mathcal{A}} \times_1 g_v(\theta_t) \times_2 \psi(g_b(\beta_t)) \tag{6.8}$$

where $\beta_t \in \mathbb{R}$ is the spline parameter (one-dimensional) and $g_b : \mathbb{R} \rightarrow \mathbb{R}^e$ is a spline fitting function which maps from the parameter space into the embedding space and satisfies $\boldsymbol{x}_t = g_b(\beta_t)$. Any combination of view manifold parameter θ_t and body configuration manifold parameter β_t can generate new image using Eq. 6.8.

For one dimensional representation of the multiple cycles, we use mean-manifold representation for parameterization. The mean-manifold is parameterized by spline fitting by a one-dimensional parameter $\beta_t \in \mathbb{R}$ and a spline fitting function $g_b : \mathbb{R} \rightarrow \mathbb{R}^e$ that satisfies $\boldsymbol{x}_t = g_b(\beta_t)$ is used to map from the parameter space into the embedding space, where e is the dimension of embedding space and $\boldsymbol{x}_t \in \mathbb{R}^e$ is the embedding coordinate. We use a three dimensional space ($e = 3$) for the embedding of the kinematic manifold in this case.

Parameterizing General Motion Manifold:

For complex motions such as aerobics, dance, etc., where the manifold dimensionality is not known, a two-dimensional embedding space is used to represent the manifold. In such case, the kernel functions centers in Eq. 6.5 are fit to the embedded manifold through fitting a Gaussian mixture model. To learn the dynamics in such case, we learn a flow field in the embedding space.

Given a sequence of N body configuration embedding coordinates on the kinematic manifold, $\boldsymbol{X} = \{\boldsymbol{x}_1 \cdots \boldsymbol{x}_N\}, \boldsymbol{x}_t \in \mathbb{R}^2$ we can directly get flow vectors, representing the velocity in the embedding space, as $v(\boldsymbol{x}_t) = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$. Given this set of flow vectors, we can estimate a smooth flow field over the whole embedding domain where the flow $v(\boldsymbol{x})$ at any point $\boldsymbol{x}$ in the space can be estimated as $v(\boldsymbol{x}) = \sum_{i=1}^{N} b_i k(\boldsymbol{x}, \boldsymbol{x}_i)$ using Gaussian kernels $k(\cdot, \cdot)$ and linear coefficients $\boldsymbol{b}_i \in \mathbb{R}^2$ which can be obtained by solving a linear system. The smooth flow field is used to estimate the how the body configuration will change in the embedding space which is used in tracking to propagate the particles. Fig. 6.5 (d) shows an example of the motion flow field for a ballet dance motion.

6.5 Experimental Results

We tested performance of our approach with different kinds of motion using synthetic data and real data. In order to learn view and configuration manifold, we used synthesized shapes rendered from real motion capture data. Typically, motion sequence from 12 discrete views are rendered to learn the view manifold. To evaluate the approach we used both synthesized and real data. synthesized data facilitates quantitative analysis of configuration and view estimation. In the experiments shown here we mainly used silhouettes to represent the observations.

However, the approach provides a generative model for contours and can easily integrated with edge based observations with proper observation model. To evaluate the 3D configuration estimation, the embedded body configuration is mapped to 3D joint angles position space through learning a RBF mapping from the embedding space to the kinematic space. The error for a given body configuration $\boldsymbol{x}_i$ is computed by average absolute distance between individual markers and the recovered 3D joint location similar to Brown HUMANEVA-I dataset [120].

6.5.1 Tracking on the Torus

Brown HUMANEVA-I dataset Evaluation

We measured 3D reconstruction error using Brown HUMANEVA-I dataset [120]. We generated synthetic training data of walking silhouette from motion capture data using animation software Poser®. 12 different views $(10^o, 30^o, \cdots, 360^o)$ are collected for walking on a circle motion. We extracted silhouettes using background subtraction. Joint locations of the validation set and of one cycle of training sequence are extracted and normalized to represent *normalized pose* which is invariant to subject's rotation and translation. We achieved this pose normalization by computing joint location after rotating each joint transformation into body centered coordinate and re-centering translation based on mean node location in each frame. We collected three subjects' validation sequences to estimate the performance of inferring 3D body pose.

We estimate body pose from the maximum a posterior (MAP) estimation of body configuration from the particle filtering. We used 900 particles ($N_\beta = 900$) in the experiment to represent view and body configuration on the torus manifold. We reconstructed 3D body pose from estimated body configuration parameter and one 3D body pose cycle from training sequence for each subject. We learned mapping between body configuration parameter and 3D body pose from the selected training sequence. After that, we can infer 3D body pose for any estimated body configuration parameter. We measured errors in estimated body pose by average absolute distance between individual markers as in [120] Fig. 6.6 (d) shows average error in each frame.

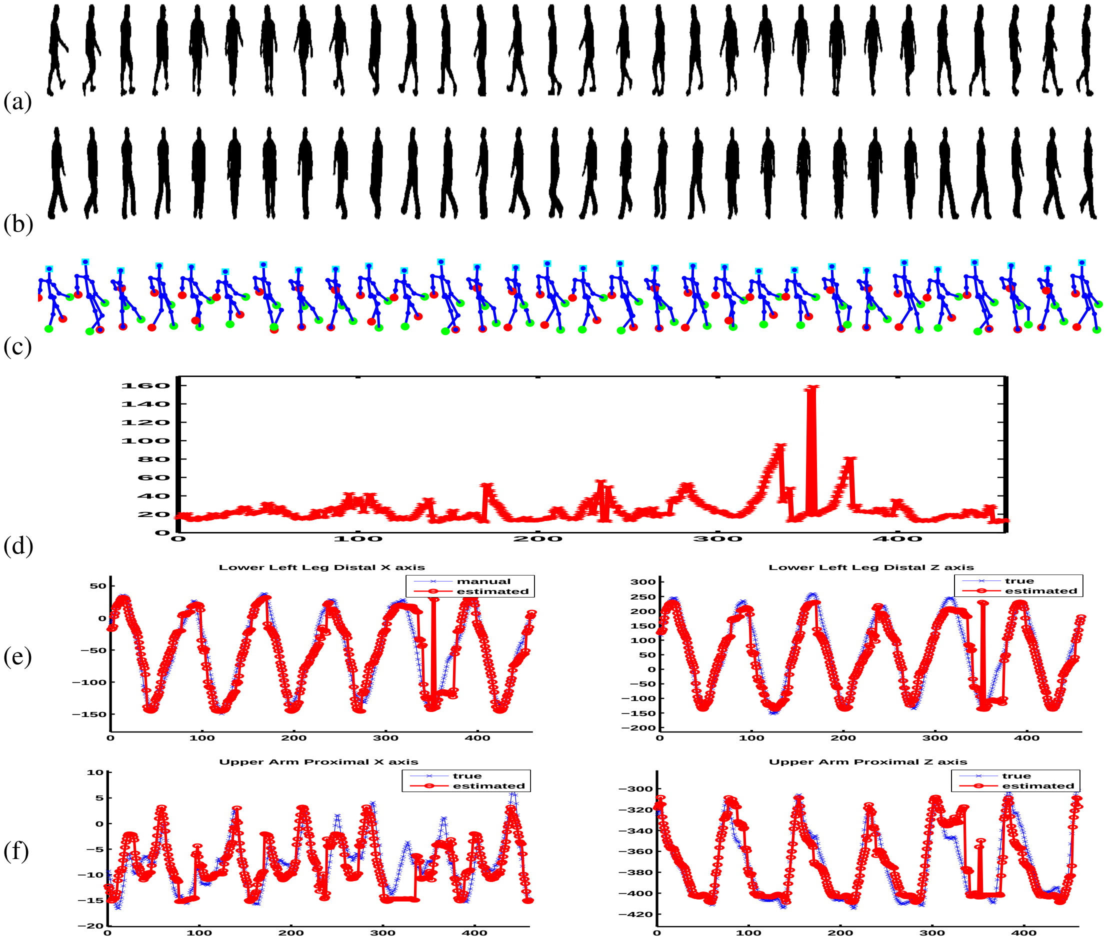


Figure 6.6: 3D body pose reconstruction using torus manifold for walking sequence (HUMANEVA-I): X-axis: frame number, Y-axis: joint location value (unit:mm). (a) Input silhouettes. (b) Reconstructed silhouette based on estimated view and body configuration. (c) Reconstructed 3D body pose. (d) Average errors in joint locations in each frame. (e)(f) True and estimated joint location x and z values for *Lower left leg distal* and *Upper right arm proximal*.

Comparison to Other Representations

We compared the torus representation with other embedding approaches for the task of body pose estimation. Since we used a torus as a two-dimensional manifold embedding for view and body configuration representation, we also used two-dimensional embedded representation obtained from LLE [111] and Isomap [127]. We used the same number of particles in the two-dimensional embedding space for all approaches. We also compared nearest neighborhood search to see the best result we can get from the collected data itself. Table 6.1 shows average error for the different approaches. For the case of nearest neighbor (NN), we searched for the nearest silhouette from training sequence and used its corresponding 3D joint location as reconstruction. Torus embedding shows much better performance than other manifold representation.

Table 6.1: Average error in normalized 3D body pose estimation in different embedding

Embedding Type	**LLE**	**Isomap**	**NN**	**Torus**
Average Error in mm	62.19	61.08	49.52	24.08

Shape Style Adaptation:

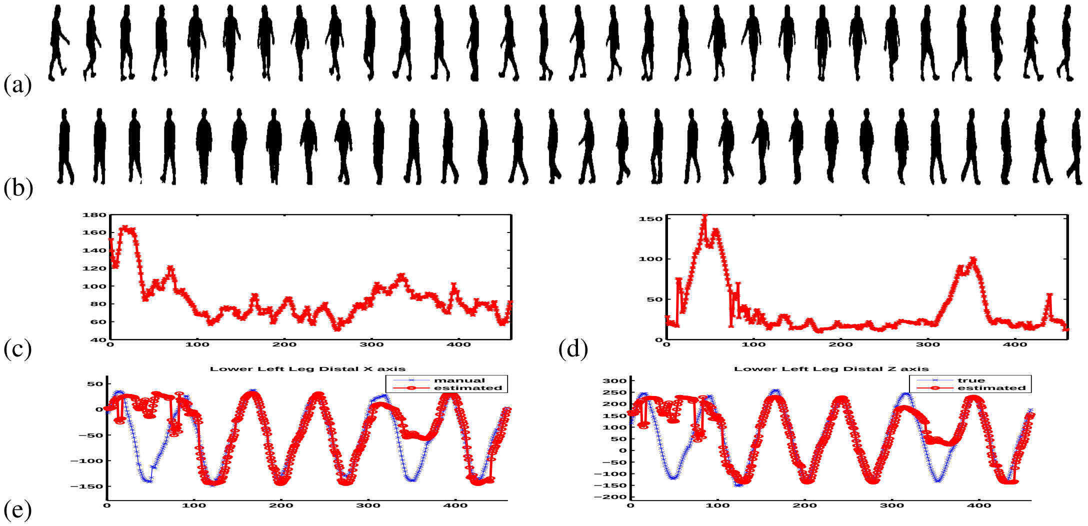


Figure 6.7: Style adaptive circular gait sequence tracking: (a) Original test silhouettes. (b) Estimated silhouettes with style adaptation. (c) Measured shape contour error in each frame in style adaptation. (d) Measured 3D reconstruction error (Average errors in joint locations in each frame in different embedding). (e) True and estimated joint location x and z values for *Lower left leg distal*

As we can model shape variations in different people as style change, we can adapt to observed shape by estimating the style factor a^s in Eq. 6.3 to explain observed shape. New person's style can be represented by combination of training person style. In our experiment, we captured people walking sequence on the treadmill with multiple camera. For our experiment, we collected sequences from 4 different people with 7 different views using synchronized camera. We started from mean style. As style adaptation goes on, the 3D reconstruction errors and 2D image reconstruction errors are decreased. Fig. 6.7 shows experiment results for the HUMANEVA-I dataset we used in the previous experiment. At the beginning, shape contour error (c) are large but it decrease as the style estimation get more accurate parameters. Similarly, the estimated 3D body pose shows decrease in error as time passes after large errors at the beginning when we used just mean style.

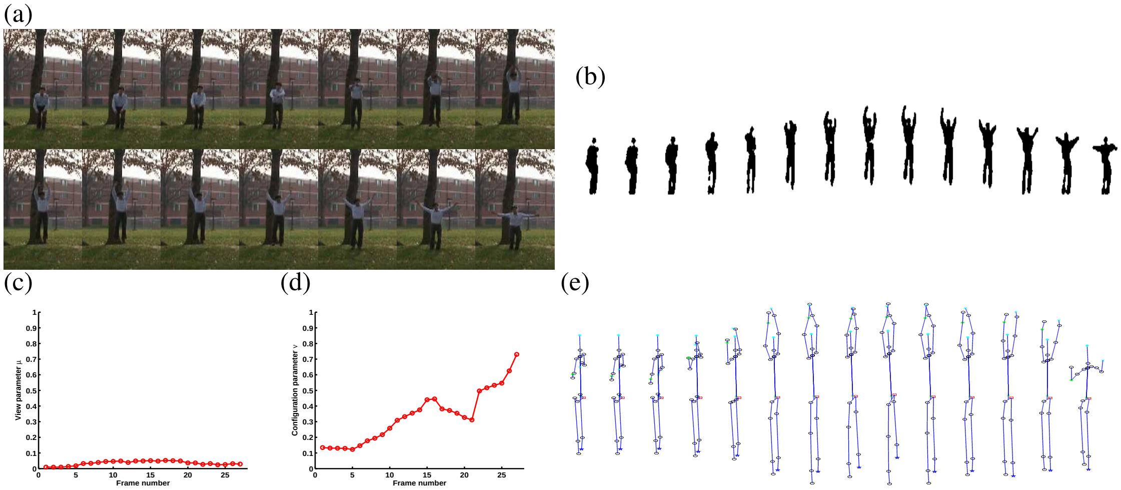


Figure 6.8: Outdoor fixed view jump motion. (a) Input image. (b)Input silhouette. (c) Estimated view. (d) Estimated body configuration. (e) 3D body pose reconstruction based on estimated body configuration.

Jump sequences

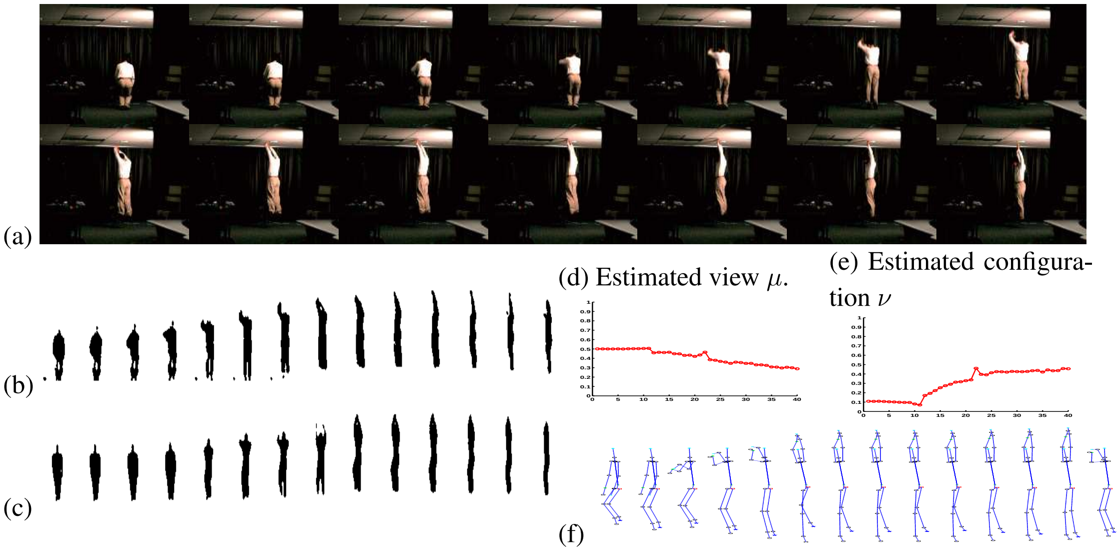
We evaluated the approach with a jump motion (example of open manifold) where the subject can rotate in the air while jumping. We used motion captured data to learn the model using geodesics-based embedding on the torus. Fig. 6.8 shows estimation of view and body configuration in outdoor environment. Despite inaccurate silhouette extraction (Fig. 6.8 (a)), our model estimate body configuration accurately (Fig. 6.8 (e)).

Fig. 6.9 (a) shows jump motion with body rotation in the air. Estimated view parameter shows constant view parameter change due to body rotation in Fig. 6.9 (d). Simultaneously the estimated configuration parameter enables reconstruction of 3D body pose (Fig. 6.9 (f)).

Edge-based Contour Tracking

We tested the approach with real data without background subtracted contours. Instead Chamfer matching is used as an observation model given edges extracted from the images.

We tested for a walking sequence along a circle with fixed camera view. Fig. 6.10 (a),(b)



(d) Estimated view μ.

(e) Estimated configuration ν

Figure 6.9: Indoor jump motion with rotation. (a) Input image. (b) Input silhouettes. (c) Reconstructed silhouettes. (d) Estimated view. (e) Estimated body configuration. (f) 3D body pose reconstruction based on estimated body configuration.

shows tracking results for walking sequence with view variation. You can see spiral motion on the torus manifold due to simultaneous change of view and body configuration. The tracking on the torus manifold can achieve reliable tracking result with prior dynamic constraints on the manifold even weak edge cues.

Golf Swing Tracking

In this experiment we tested tracking performance of golf swing from unknown camera and view. In this experiment, we can recover correct view and body configuration. Fig. 6.10 (c),(d) shows tracking results. Since the view is unknown, we start from a uniform distribution, i.e., the particles are spread along the big circle on the torus (the same μ) at the beginning and it converged to one area.

6.5.2 Estimation Using View Manifold and One-Dimensional Motion Manifold

Brown HUMANEVA-I dataset - Walking in a Elliptical Path

We tested 3D body posture estimation accuracy using Brown HUMANEVA-I dataset [120], which provides ground truth data for 3D joint locations for different types of motions. We used a circular walking sequence, which has continuous relative view variations between the subject

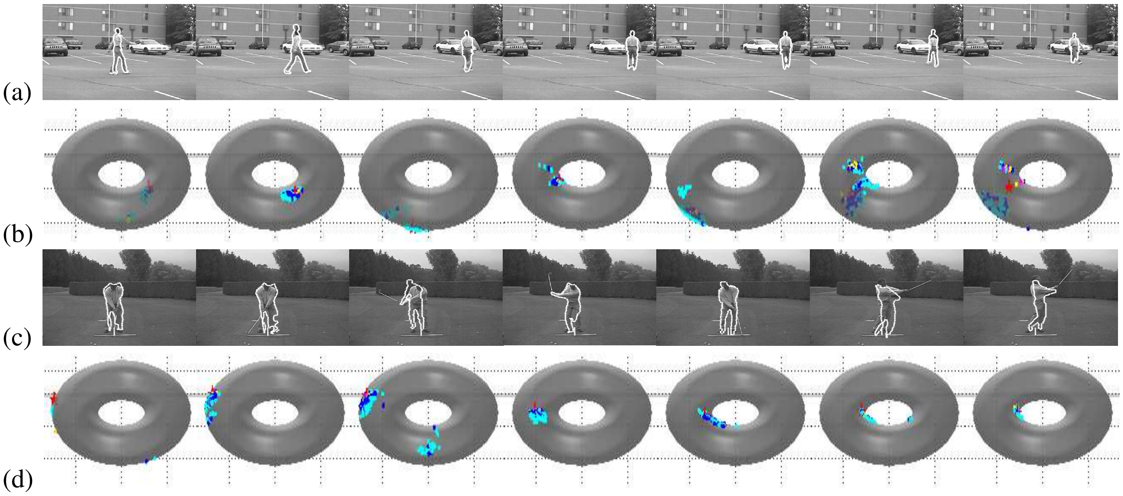


Figure 6.10: Circular gait sequence tracking: (a) Estimated shape contours. (b) View and configuration particle distributions on torus manifold. Golf swing tracking: (c) Estimated shape contours (d) View and configuration particle distributions on torus manifold.

and the camera. We normalized original joint location in HUMANEVA-I dataset into body-centered coordinate system, which is invariant to subject's body rotation and translation. For the estimation of 3D body pose, we selected one cycle from the training sequence to learn mapping from embedded kinematic manifold to the 3D kinematic space. Fig. 6.11 shows estimated view, body configuration, and corresponding 3D body pose reconstruction. The estimated parameters fit very well a constant speed linear dynamic system for both the configuration and view parameters. The average error in HUMANEVA-I dataset for 512 frames in Subject *S1* is 26.294 mm.

Golf swing - one dimensional open manifold: Golf swing is a complicated and non-periodic motion. Still it can be parameterized by one-dimensional kinematic manifold. The motion manifold is *open* as the final body pose is very different from the initial motion (Fig. 6.12 (h)). We learned view manifold after synthesizing 12 discrete views from motion capture data. For the test sequence, we controlled camera location in a constant speed to move in a 360^o circular trajectory during the golf swing motion. Fig. 6.12 (e) shows estimated joint pdf using particle for body configuration β and view configuration θ. Estimated view in Fig. 6.12 (g) correctly reflects constant change of view.

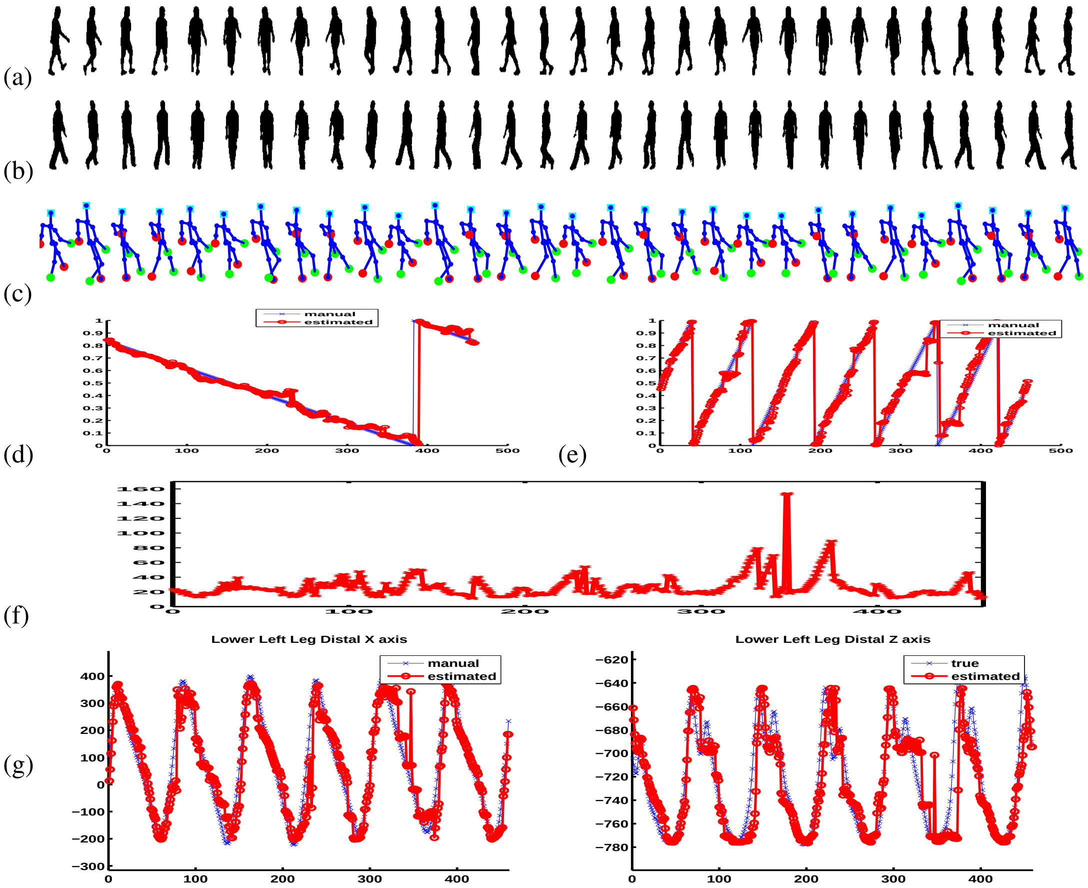


Figure 6.11: Estimation using view manifold and kinematic manifold for walking sequence (HUMANEVA-I): (a) Input silhouettes. (b) Synthesized silhouettes after view and body configuration estimation. (c) Reconstructed 3D body pose. (d) Estimated view parameters. (e) Estimated body configuration parameters. (f) Joint location error in each frame. (g) The comparison of joint location between estimated value and ground truth value :x and z values for *Lower left leg distal.*

Basketball pass motion

The basketball pass action similar to many other sport activities can be embedded on a one dimensional motion manifold when we count single cycle. Because of many camera motion and human body rotation in arbitrary view in sport video sequence, modeling actions in arbitrary view is crucial in general sport activity tracking and recognition. In our experiment,we can reliably estimate change of body configuration in spite of noisy silhouette inputs as shown in Fig. 6.13 (d).

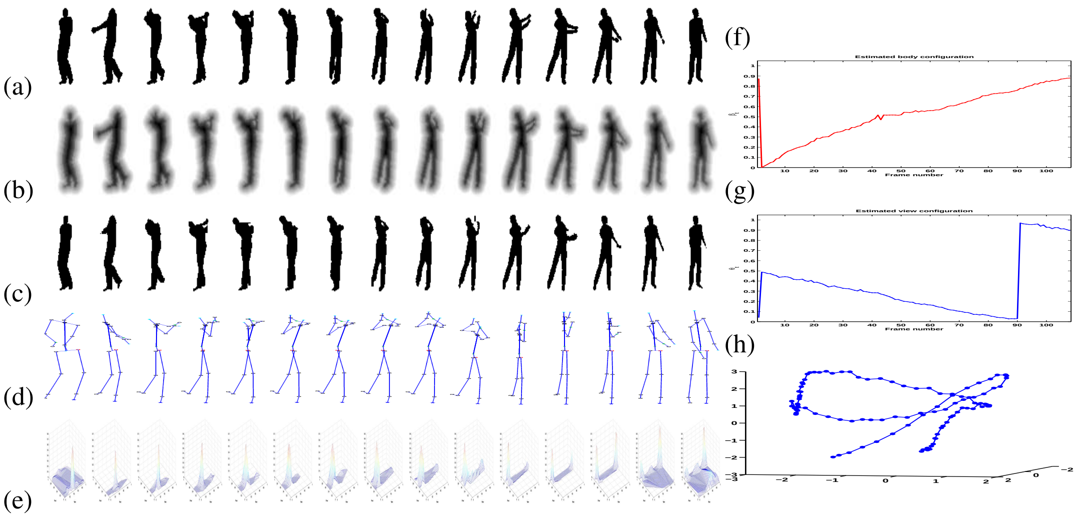


Figure 6.12: Golf swing: (a) Input silhouette sequences. (b) Implicit shape representation of input silhouettes. (c) Synthesized silhouettes for estimated body pose and view. (d) Reconstructed 3D body pose. (e) Estimated probability densities for view and body configuration parameter space. (f) Estimated body configuration. (g) Estimated view configuration. (h) Kinematics manifold of the golf swing sequence that we used.

6.5.3 Estimation From General Motion Manifolds

Simple sport motions like ball passing, catch/throw can not be parameterized by a one-dimensional manifold due to variability in body configuration when the motion is repeated. For example, when we catch and throw a ball repeatedly in the air, the catch action changes according to the falling ball locations. Moreover, many activities like dancing, aerobics are high dimensional in their kinematic manifold.

Catch/throw motion

We used catch and throw sequences with variations of motion in each catch and throw cycles, which is represented as different trajectories in the body configuration embedding. Fig. 6.14 (d) shows the embedding space for configuration. Fig. 6.14 (e) shows the obtained posture-invariant view manifold. Fig. 6.14 (f) shows estimated view for the test sequence shown in Fig. 6.14 (b) which exhibits camera motion with constant speed. Fig. 6.14 (g) shows the flow field for such motion. Recovered body configuration is shown in Fig. 6.14 (c) as reconstructed silhouettes from recovered configuration and view parameters.

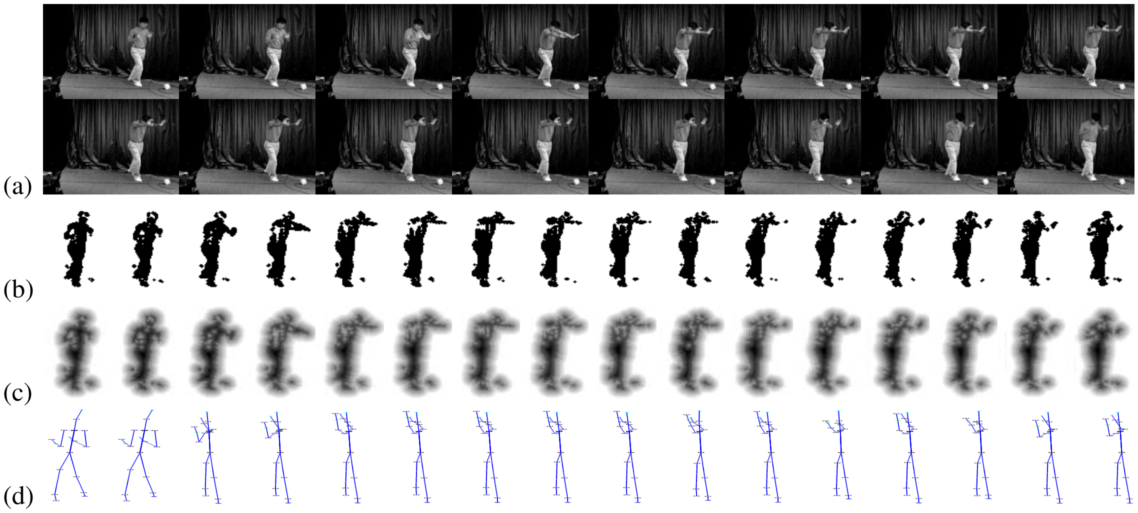


Figure 6.13: Basketball pass: (a) Captured images (frame number: $6, 12, \cdots, 96$). (b)(c) Extracted silhouettes and corresponding implicit shape representations. (d) Reconstructed 3D body pose based on estimated configuration parameters

Aerobic Dancing Sequence

We used locally linear embedding (LLE) [111] to learn nonlinear manifold embedding for the dancing sequence. Two dimensional manifold embedding is used to represent repetitive dancing sequence as in Fig 6.15 (b). Then we learned view-dependent dynamic shape contour models from 12 synthetic view. We modeled view manifold after decomposition of the view dependent nonlinear mapping. Fig. 6.15 (c) shows the learned posture-invariant view manifold. We tested performance of view and body pose estimation using synthetic rendered data with two different view settings. For the first test sequence, we used a fixed intermediate view, which is not used for the view learning. In our training view, we collected view from $0^o, 30^o, 60^o, \cdots, 330^o$ and We used 45^o views in our test experiment. The other sequence has continuous view variations from 0^o to 90^o view. Average joint location error in each frame is shown in Fig 6.16 (d),(h). We used 30 particles for view estimation and 30 particles for body pose estimation. Even though these small number of particles like 30 for view ,experimental results show reliable estimation of view and body configuration in spite of view variations after initial stage.

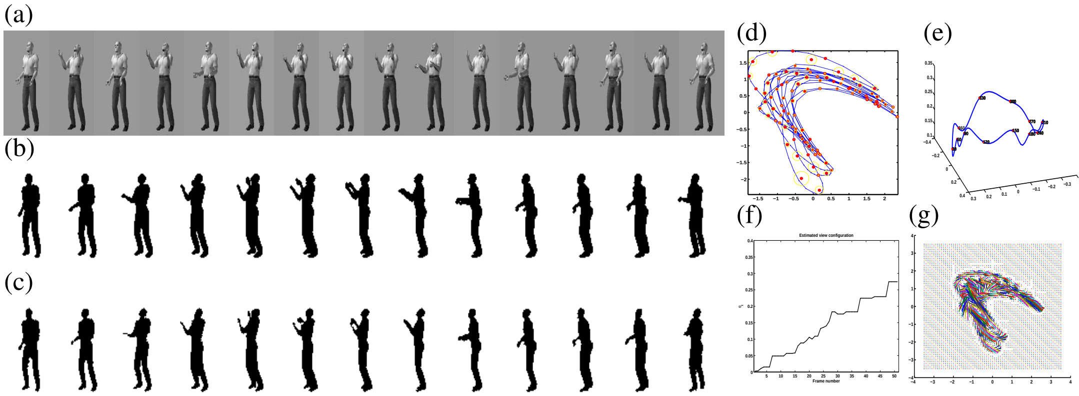


Figure 6.14: Catch/throw motion (Evaluation): (a) Rendered image sequence of catch/throw motion capture data (frame 3, 25, 47, 69, $\cdots$, 333) (b)Test sequence with a moving camera and corresponding rendered action. (c) Estimated shape sequence after view and configuration estimation. (d) Two-dimensional motion manifold embedding and selected kernel points. (e) Posture-invariant view manifold of the catch/throw motion in 3D space (f) Estimated view configuration. (g) flow field interpolation on the embedding space.

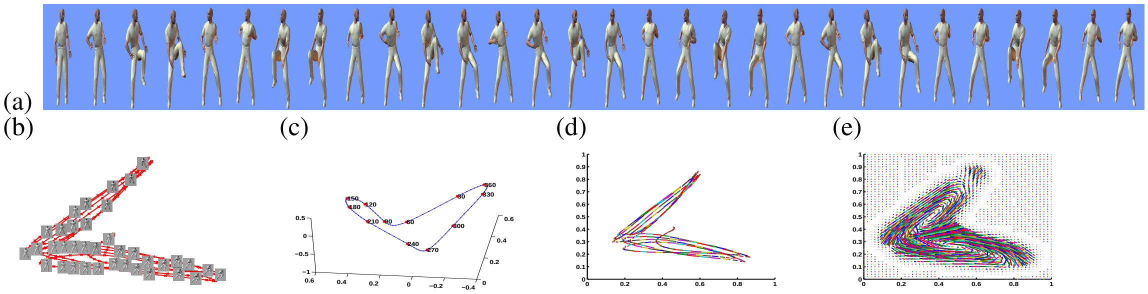


Figure 6.15: Dancing sequence : (a) Rendered image sequence of dancing data. (b) Body configuration embedding for dance sequences. (c) View manifold for dancing sequence. (d)(e) Velocity field and its interpolation.

Ballet Motion

Ballet motion has frequent body rotation and the motion is very complicated since arms and legs are moving independently. However, the motion is still constrained by the physical dynamics in the motion and the rules in the ballet dancing. Fig. 6.17 (a),(b),(c) show results for estimated configuration and view for the ballet motion which was shown in Fig. 6.2 and Fig 6.5.

6.6 Summary

We formulated view variant human motion tracking as tracking on a torus surface. We use the torus as a state space for both body configuration and view. We learn how the torus deform

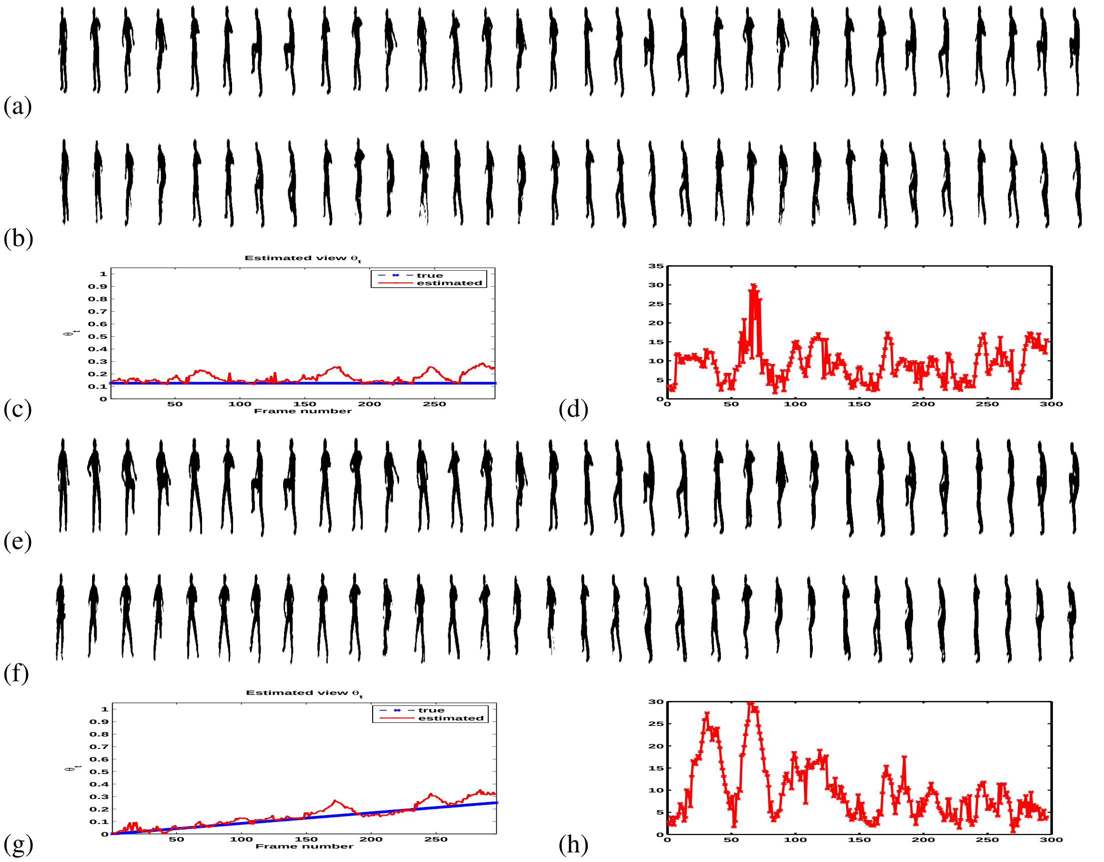


Figure 6.16: Dancing sequences evaluation: (a) Input silhouettes for test from a fixed view. (b) Reconstructed silhouettes for fixed view. (c) Estimated view parameters for fixed view. (d) Average joint location error in fixed view. (e) Silhouettes from rotating view. (f) Reconstructed silhouettes from rotating view. (g) Estimated view parameters for rotation view. (h) Average joint location error in rotating view.

to the actual visual manifold and to the kinematic manifold through two nonlinear mapping functions. The torus model is suitable for one dimensional manifold motions, whether periodic, as walking, running, etc., or non periodic, as golf swings, jumping, etc. The experimental results showed that such model is superior than other representations for the task of tracking and pose/view recovery since it provides a low dimensional, continuous, uniformly spaced state representation. We also show, how the model can be generalized to the full view sphere and how to adapt to different people shapes.

In addition, we introduced an approach for explicit modeling of body configuration and view in two separate low dimensional embedded representations. The body configuration is embedded from kinematic data, i.e., invariant of the view. The view can represented in a

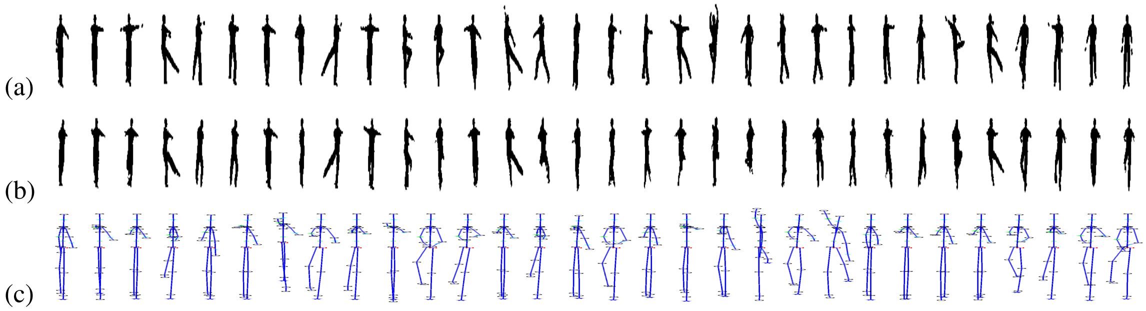


Figure 6.17: Ballet motion evaluation: (a) Test silhouette sequences. (b) Synthesis of silhouettes based on estimated body pose and view. (c) Reconstruction of 3D body pose based on estimated body configuration *Shown in a body-centered coordinate- without body rotation*.

posture-invariant manner. As a result, we have a generative model that parameterize the motion, the view, and the shape style. The model is appropriate for tracking and pose estimation of complex motion from un-calibrated stationary or moving camera. We showed several experimental results and quantitative evaluations for wide varieties of motion including simple motion such as gait and gold swing to complex motion such as aerobics and ballet dancing. The model can successfully self initialize, track, and recover the parameters for view and 3D configurations even with a moving camera.

Chapter 7

Facial Expression Analysis and Synthesis

Facial expression passes through nonlinear shape and appearance deformations with variations in different people and expressions. We present nonlinear shape and appearance models for facial expression analysis and synthesis using nonlinear generative models for different facial expressions in different people. To achieve accurate shape normalized appearance models, we utilize nonlinear warping using thin plate spline (TPS). A novel nonlinear generative model using conceptual manifold embedding and empirical kernel maps for facial expressions provides facial shape and appearance samples according to the configuration, personal style, and expression parameters. We can recognize facial expressions based on estimated facial expression parameters after iterative estimations of facial expression and style. In addition, the model provides accurate synthesis of facial expression sequences even with high nonlinear deformations of shape and appearance during facial expressions. In addition, we combine the global nonlinear appearance estimation and direct local fitting for tracking large facial deformations. The generative model allows the Bayesian tracking of facial expressions using particle filter and simultaneous estimation of the expression types. Based on global shape model, we estimate global transformation and global deformation that provides a normalized appearance template. Local fine fitting is achieved by shape parameter estimation based on direct thin plate spline warping parameter estimation using the normalized appearance template from the global nonlinear appearance model.

7.1 Overview:Modeling Subtle Facial Motions

People get more interests in modeling and analyzing dynamic human motions for natural human computer interaction, surveillance system in compute vision, and animation of human motions for films or computer games. Recently, demands for accurate modeling and analysis of facial expressions are growing with new applications of facial motion analysis like deception detection and affective computing. Recognition of emotional states and synthesis of facial expressions are one of the key components for intelligent affective human computer interactions [102]. However, it is difficult to model subtle facial motions with current linear subspace-based models as facial motions pass through nonlinear shape and appearance deformations with variations in different people and expressions.

Most of current systems for analysis and tracking of facial motions from shape and appearances are based on linear models. Active shape models (ASM) [30] are well known linear models for facial motion analysis and tracking using point distribution models in linear subspace. By constraining deformation of point distributions into a linear subspace of the training shapes during local search, it achieves robust fitting of face models [30]. Active appearance models (AAM) combine the linear shape model of points distributions in ASM and the linear texture appearance model by aligning appearance models into a normalized shape using piecewise affine warping [29]. Iterative model refinement algorithms are proposed based on a prediction model, which is learned as a regression model using observations of linear shape and appearance model parameter variations after perturbations.

Bilinear and multilinear models are applied to improve accuracy in modeling facial expression recognitions [1, 140] and facial expression synthesis [24]. All these models are based on linear shape and appearance models with extensions for multiple factors. In addition, any of these models do not count dynamics in facial expressions except [77]. Feature-based facial expression recognition system may overcome the limitation of linear subspace of templates by directly tracking features like facial action units [83]. Feature based approaches, however, are hard to model appearance variations in facial expression and may not be able to synthesize appearance of new facial expressions.

We propose nonlinear shape and appearance models for facial expression analysis and synthesis. When dealing with dynamic facial expressions, image sequences lie on low dimensional nonlinear manifolds embedded in a high dimensional input space. We model facial expressions by explicit modeling of configuration manifolds and decomposing variability due to different people and expression types. Nonlinear generative models using low-dimensional conceptual manifold embedding and empirical kernel mapping are developed to learn nonlinear shape and appearance model in low dimensional spaces. This generative model provides a global shape and appearance deformation model during facial expressions. Iterative estimation of the model parameters allows recognition of facial expressions for a given image sequence. To achieve accurate shape-normalized appearance images for learning our models, we employed thin-plate spline (TPS) warping.

We combine global nonlinear appearance tracking and local fitting of shape models directly from adaptive appearance templates in order to achieve tracking of large facial shape and appearance deformations. Tracking provides estimated states of global transformation and deformation based on our generative model and particle filter within the Bayesian framework. However, it fails to capture accurately and estimated global transformations are sometimes insensitive to small misalignments and local deformations. When we apply local deformation for a given template, we can achieve accurate alignment and local fitting for small facial deformations for a given template. The estimation of local deformation based on a single template, however, fails to track large facial motion deformations. We used estimated global transformation and deformations as initial state for local fitting. The global model, which supports large shape deformations, provides shape normalized appearance models for local fitting. By combining the global appearance model and local fitting, we can achieve accurate estimations of facial motions in large deformations.

7.2 Dynamic Shape and Appearance Models for Facial Expressions

To develop a nonlinear generative shape and appearance model, we extract normalized shape and appearance models by shape alignment and nonlinear warping of dynamic appearance to mean shape using TPS. The normalized shape vector and the appearance vector are combined as a new shape and appearance vector for the nonlinear generative model. We model facial motions as configuration variations in time with different facial shape and appearance deformations according to given expression type.

7.2.1 Shape-Normalized Appearance Models Using Thin-plate Spline Warping

To develop nonlinear generative shape and appearance models, we first represent shapes by distributions similar to ASM of landmark points and compute a mean shape that is used for appearance normalization. We describe the ith shape by n landmark points as a vector $\boldsymbol{p}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \cdots, x_{in}, y_{in})$. From collected landmark points of different people with different expressions, we compute the mean shape after shape alignments by weighted similarity transformation [30]. As some components, like a nose, are more reliable than other face components, like a mouse contour, in facial expressions, we weighted each landmark based on the reliability of the face components. By shape normalization, appearance vectors will establish good correspondences between each element of the shape-normalized appearance vectors and, therefore, meaningful algebraic operations between appearance vectors can be achieved.

To achieve shape normalization, piecewise-affine warping is frequently used in linear appearance models [125, 89]. However, piecewise-affine warping can cause artifacts around boundaries for non-rigid shape deformation due to facial motions [28]. We use thin plate spline (TPS) warping for the non-rigid registration of appearance images to the mean shape. TPS warping have been widely used in medical image alignments and non-rigid deformations [108, 26] after popularization by Bookstein [16]. Given an image sequence $\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_{N_K}$, where N_K is the number of image frames, with corresponding shape vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_K}$, we can obtain the mean shape $\boldsymbol{p}_0$. We need to warp every image $\boldsymbol{I}_j$ with shape vector $\boldsymbol{p}_j$ into new appearance image by shape landmark points using TPS.

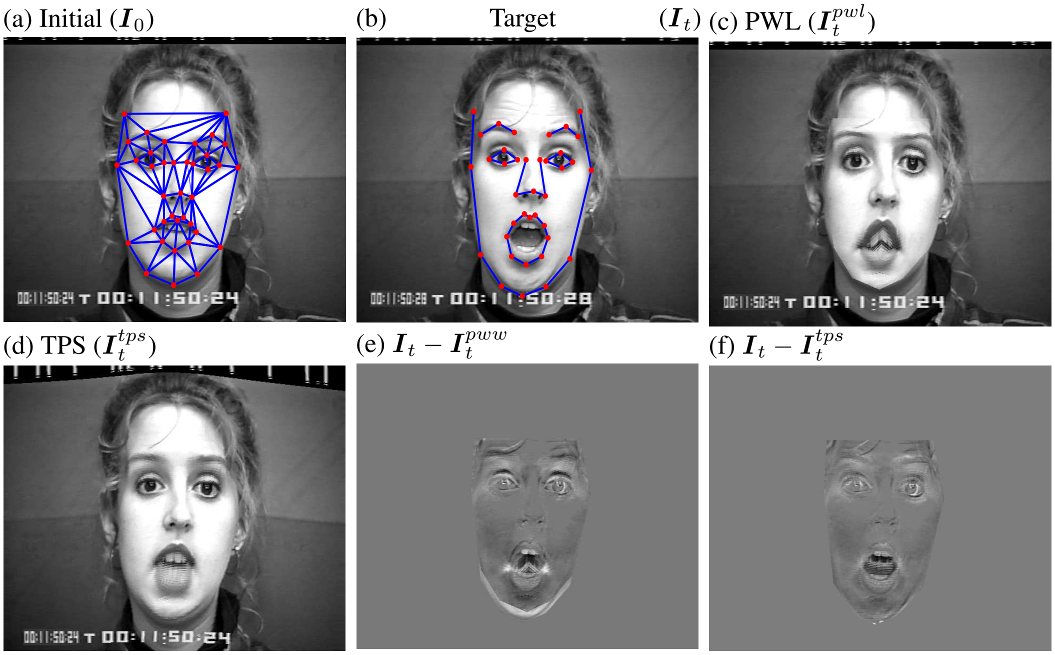
TPS warping leads to smooth deformations and accurate normalization of non-rigid deformations of appearances in facial expressions. In case of backward warping we need to warp output image coordinates into input image coordinates by TPS warping from $\boldsymbol{p}_0$ to $\boldsymbol{p}_j$ and interpolate intensity values based on the warped coordinate. TPS warping specifies a mapping in the form

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{2n} w_i U(|\boldsymbol{p}_0 - (x, y)|) \tag{7.1}$$

which minimizes bending energy

$$E_f = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dx dy$$

where $U(x, y) = r^2 log r^2, r = \sqrt{x^2 + y^2}$. When we perform backward warping from the



(a) Initial ($\boldsymbol{I}_0$) (b) Target ($\boldsymbol{I}_t$) (c) PWL ($\boldsymbol{I}_t^{pwl}$)

(d) TPS ($\boldsymbol{I}_t^{tps}$) (e) $\boldsymbol{I}_t - \boldsymbol{I}_t^{pww}$ (f) $\boldsymbol{I}_t - \boldsymbol{I}_t^{tps}$

00:11:50:24 T 00:11:50:24
00:11:50:28 T 00:11:50:28
00:11:50:24 T 00:11:50:24
00:11:50:24 T 00:11:50:24

Figure 7.1: Image warping: (a) The original image with shape landmarks. It also shows delaunay triangulation [4] result. (b) The target image with its shape landmarks. (c) Piecewise linear affine transformation based on shape triangulation for shape landmark points. (d) TPS warping based on shape landmark correspondences. (e) (f) Image difference between target image and warped image by piecewise warping (PWL) and by TPS warping (TPS)

shape $\boldsymbol{p}_0$ to a shape $\boldsymbol{p}_j$, the coefficients in the mapping function Eq. 7.1 can be computed as follows:

$$(w_1 \cdots w_{2n} | a_1 a_x a_y)^\mathsf{T} = \boldsymbol{L}^{-1}\boldsymbol{Y}, \tag{7.2}$$

where $\boldsymbol{Y} = \begin{bmatrix} x_{j1} & x_{j2} & \cdots & x_{jn} \\ y_{j1} & y_{j2} & \cdots & y_{jn} \end{bmatrix}$, $\boldsymbol{L} = \left[\begin{array}{c|c} \boldsymbol{K} & \boldsymbol{P} \\ \hline \boldsymbol{P}^\mathsf{T} & \boldsymbol{O} \end{array}\right]$, $\boldsymbol{K}_{ij} = U(||(x_{0i}, y_{0i}) - (x_{0j}, y_{0j}||)$ and the i-th row of $\boldsymbol{P}$ is $(1, x_{0i}, y_{0i})$. Computing the coefficients, we need to compute the inverse of matrix $\boldsymbol{L}$ only once as it only depends on $\boldsymbol{p}_0$. Fig. 7.1 (e) and (f) shows intensity differences between target image (b) and warped images (c) and (d) within target shape contour. When we measure errors between true target image appearance and warped ones by pixel intensity differences, the TPS warped appearance shows 20% less errors than the piecewise linear one as its warping based on corresponding landmark points describe nonlinear deformation of appearance more accurately in large shape deformation during facial expressions. The piecewise linear warping result may be improved using more dense landmark points and elaborate triangulation.

Given an image sequence, the kth image $\boldsymbol{I}_k$ is represented by the aligned shape $\boldsymbol{p}_k$ and shape-normalized appearance $\boldsymbol{a}_k$, where the appearance vector, $\boldsymbol{a}_k$, is the vector representation of pixels which are inside the mean shape contour after warping the image to the mean shape by TPS, i.e.,

$$\boldsymbol{a}_k = \underset{\boldsymbol{\xi} \in \boldsymbol{p}_0}{\boldsymbol{I}_k} \quad (W_{\boldsymbol{p}_0}(\boldsymbol{\xi}, \boldsymbol{p}_k)). \tag{7.3}$$

We combine the shape vector and the appearance vector as a new vector $\boldsymbol{y}_k = [\boldsymbol{p}_k^\mathsf{T}\ \boldsymbol{a}_k^\mathsf{T}]^\mathsf{T}$ for the facial expression analysis. In this case, the dimension of $\boldsymbol{y}$ is $N_{as} = 2n + N_a$ where the number of pixels within the mean shape is N_a. The combined shape and appearance vectors are used in modeling facial motions using the nonlinear generative model as will be described in the next section.

7.2.2 Nonlinear generative models with manifold embedding

Nonlinear dimensionality reduction has been recently exploited to model manifold structures in face recognition [121] and facial expression analysis [22]. When unsupervised data-driven manifold embedding techniques are used, resulting embedded manifolds of the same type of facial expressions performed by different people will be quite different and it is hard to find a unified representation of the manifolds. But, conceptually all these manifolds are the same 1-dimensional circular curves for expressions, which move from neutral expressions to target expressions and return back to neutral expressions. Using this conceptual manifold embedding and nonlinear mapping, we can model dynamics of facial expressions in a low dimensional space.

A set of image sequences which represent full cycles of the facial expressions are used for conceptual embedding of facial expressions using a unit circle. The image sequences are not necessarily to be of the same length. We denote each sequence by $Y^{se} = \{y_1^{se} \cdots y_{N_{se}}^{se}\}$ where e denotes the expressions and s is person's identity. Let N_e and N_s denote the number of expressions and number of people in the training data respectively. Each sequence is temporally embedded at equidistance on a unit circle such that $x_i^{se} = [cos(2\pi i/N_{se})\ \ sin(2\pi i/N_{se})], i = 1 \cdots N_{se}$.

Given a set of distinctive representative points on the unit circle $\{\boldsymbol{z}_i \in \mathbb{R}^2, i = 1 \cdots N\}$, we can define an empirical kernel map[113] as $\psi_N(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}^N$ where

$$\psi_N(\boldsymbol{x}) = [\phi(\boldsymbol{x}, \boldsymbol{z}_1), \cdots, \phi(\boldsymbol{x}, \boldsymbol{z}_N)]^{\mathsf{T}}, \tag{7.4}$$

given a kernel function $\phi(\cdot)$. For each input sequence $\boldsymbol{Y}^{se}$ and its embedding $\boldsymbol{X}^{se}$, we can learn a nonlinear mapping function $f^{se}(\boldsymbol{x})$ that satisfies $f^{se}(\boldsymbol{x}_i) = \boldsymbol{y}_i, i = 1 \cdots N_{se}$ and minimizes a regularized risk criteria. The whole mapping can be written as

$$f^{se}(x) = \boldsymbol{B}^{se} \cdot \psi(\boldsymbol{x}) \tag{7.5}$$

where $\boldsymbol{B}$ is a $d \times N$ coefficient matrix. We have d simultaneous interpolation functions each from 2D to 1D. The mapping coefficients can be obtained by solving the linear system $[\boldsymbol{y}_1^{se} \cdots \boldsymbol{y}_{N_{se}}^{se}] = \boldsymbol{B}^{se}[\psi(\boldsymbol{x}_1^{se}) \cdots \psi(\boldsymbol{x}_{N_{se}}^{se})]$. Using these nonlinear mappings, we can capture nonlinearity of facial expression in different people and expressions.

The nonlinear mappings are different for different people and for different expressions. Higher-order singular value decomposition (HOSVD) is applied to decompose the mapping coefficients into multiple orthogonal factors as described in Sec. 4.3. The coefficient tensor is then decomposed as

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{S} \times_2 \boldsymbol{E} \times_3 \boldsymbol{F}, \tag{7.6}$$

where $\boldsymbol{\mathcal{Z}}$ is a core tensor, with dimensionality $N_s \times N_e \times N_c$ which governs interactions among different mode basis matrices, $\boldsymbol{\mathcal{B}}$is an order-three facial expression coefficient tensor, $\boldsymbol{S}$, $\boldsymbol{E}$, and $\boldsymbol{F}$, representing the basis for people, expressions and pixels respectively.

Given this decomposition and given any N_s dimensional person face vector $\boldsymbol{s}$ and any N_e dimensional expression vector $\boldsymbol{e}$, we can generate coefficient matrix $\boldsymbol{B}^{se}$ by unstacking the vector $\boldsymbol{b}^{se}$ obtained by tensor product $\boldsymbol{b}^{se} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{e}$. This can be expressed abstractly also in the generative form by arranging the tensor $\boldsymbol{\mathcal{Z}}$ into an order-four tensor $\boldsymbol{\mathcal{C}}$

$$\boldsymbol{y}_t^{se} = \boldsymbol{\mathcal{C}} \times \boldsymbol{s} \times \boldsymbol{e} \times \psi(\boldsymbol{x}_t) \ , \tag{7.7}$$

where dimensionality of core tensor $\boldsymbol{\mathcal{C}}$ is $d \times N_s \times N_e \times N$. The result of the tensor multiplication $\boldsymbol{\mathcal{C}} \times \boldsymbol{s} \times \boldsymbol{e}$ is a reconstruction of the coefficient matrix $\mathbf{B}^{se}$. We can analyze facial expression image sequences by estimating the state parameters $\boldsymbol{s}, \boldsymbol{e}$, and $\boldsymbol{x}_t$.

7.3 Facial Expression Recognition and Synthesis

7.3.1 Facial Expression Recognition

We can recognize facial expressions by the estimated expression vector $\boldsymbol{e}$. Given an input image $\boldsymbol{y}$, we need to estimate configuration $\boldsymbol{x}$, expression parameter $\boldsymbol{e}$, and personal face parameter $\boldsymbol{s}$ which minimize the reconstruction error

$$E(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}) = || \ \boldsymbol{y} - \mathcal{C} \times_1 \boldsymbol{s} \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x}) \ || \tag{7.8}$$

We assume an expression vector for a given image can be written as a linear combination of expression class vectors in the training data, i.e., we need to solve for linear regression weights α such that $\boldsymbol{e} = \sum_{k=1}^{K_e} \alpha_k \boldsymbol{e}^k$ where each $\boldsymbol{e}^k$ is one of expression class vectors in the training data. Similarly for the personal face, we need to solve for weights β such that $\boldsymbol{s} = \sum_{k=1}^{K_s} \beta_k \boldsymbol{s}^k$ where each $\boldsymbol{s}^k$ is one of K_s face class vectors.

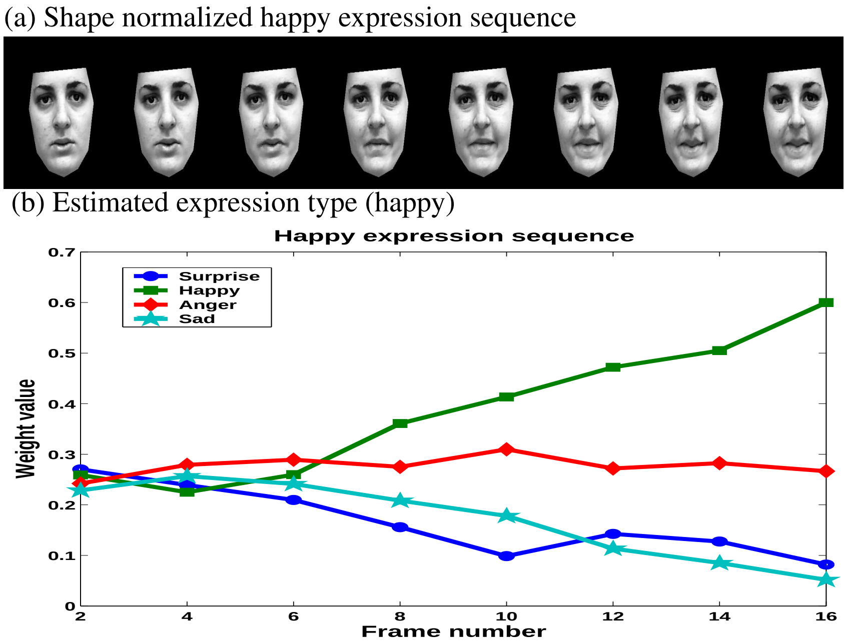
If the expression vector and the person face vector are known, then Eq. 7.8 is reduced to a nonlinear 1-dimensional search problem for configuration $\boldsymbol{x}$ on the unit circle that minimizes the error. On the other hand, if the configuration vector and the person face vector are known, we can obtain expression conditional class probabilities $p(\boldsymbol{e}^k|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$ which is proportional to observation likelihood $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}^k)$. Such likelihood can be estimated assuming a Gaussian density centered around $\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x})$, i.e.,

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{s}, \boldsymbol{e}^k) \approx N(\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(x), \Sigma^{\boldsymbol{e}^k}).$$

Given expression class probabilities, we can set the weights to $\alpha_k = p(\boldsymbol{e}^k \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{s})$. Similarly, if the configuration vector and the expression vector are known, we can obtain face class weights by evaluating image likelihood given each face class $\boldsymbol{s}^k$ assuming a Gaussian density centered at $\mathcal{C} \times_1 \boldsymbol{s}^k \times_2 \boldsymbol{e} \times_3 \psi(\boldsymbol{x})$. This setting favors an iterative procedure for solving for $\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{s}$. We applied deterministic annealing procedure as explained in Sec. 4.5.

Frame-based Facial Expression Recognition

For any given image frame, we can estimate facial expression parameter using iterative estimation of expression, style and configuration parameters. Estimated expression weights can



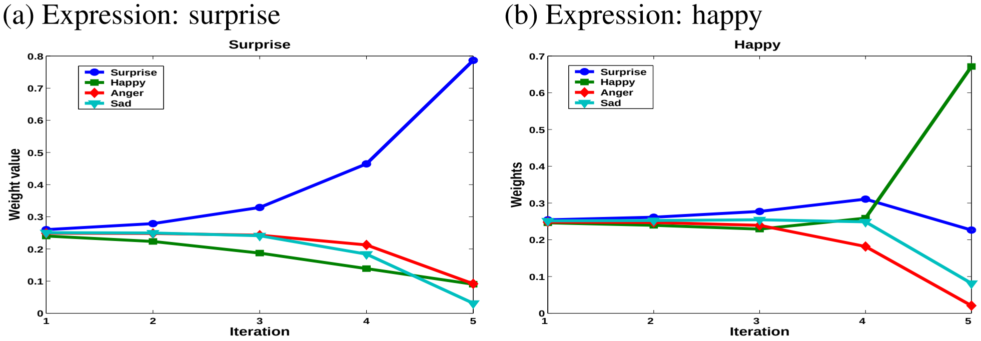
(a) Shape normalized happy expression sequence

(b) Estimated expression type (happy)

Figure 7.2: Frame based facial expression estimation: (a) Shape normalized happy expression sequence. (b) Estimated weights of expression class types.

be used directly to recognize facial expression type. Most of the facial expression recognition systems use a peak expression image for learning and recognition of facial expression. In our model, all the frames can be used for the estimation of facial expression. In addition, estimated configuration parameter shows how close to peak expression from embedding coordinate. Fig. 7.2 shows an example facial expression sequence of shape normalized appearance and its estimated weights of expression classes in every frame of the sequence. As the expression starts from a neutral expression, which is similar in all different expressions in the database [66], the estimated expression weights are similar to all four expression classes at the beginning. As expression sequence progresses to a peak expression configuration in the sequence, the estimated expression weights become discriminative. From sequence of an expression, we may recognize expression more robustly by selecting majority of recognized expression. However, we used peak expression image in testing facial expression classification in order to be comparable to other recognition systems using only peak expression frames in facial expression recognition.

The facial expression estimation in peak expression becomes an iterative estimation of expression vector and style vector as we know the configuration, the low dimensional embedding.


(a) Expression: surprise

(b) Expression: happy

Figure 7.3: Expression class weight changes in iterations:(a) True expression is surprise. (b) True expression is happy. Here the highest weight changed during iteration as style estimations also changed from mean style to specific style.

We can embed the peak expression at the opposite location from the neutral (initial) expression in the unit circle embedding. So, we don't need to estimate configuration by full search in the embedding space. The facial expression recognition becomes finding the closest expression class with the estimated expression parameter. Fig. 7.3 shows typical examples of expression weight changes through the iterations. Sec. 7.5.1 shows experimental results using peak expression image from CMU AU coded facial expression database [66].

7.3.2 Facial Expression Synthesis

Our model can synthesize facial expressions by combinations of facial expression parameters and personal face parameters. As we have decomposed the mapping space that captures nonlinear deformation in facial expressions, the linear interpolation of the face style and facial expression still captures nonlinearity in the facial expression. A new personal face vector and a new facial expression vector can be generated by a linear combination of existing personal face class vectors and expression class vectors using parameter α_i as follows:

$$\boldsymbol{e}^{new} = \alpha_1 \boldsymbol{e}_1 + \alpha_2 \boldsymbol{e}_2 + \cdots + \alpha_{N_e} \boldsymbol{e}_{N_e} \tag{7.9}$$

where $\sum_i \alpha_i = 1$. We can synthesize new personal face $\boldsymbol{s}^{new}$ similarly. A new facial expression image can be generated using new personal style and expression parameters.

$$\boldsymbol{y}_t^{new} = \mathcal{C} \times_1 \boldsymbol{s}_t^{new} \times_2 \boldsymbol{e}_t^{new} \times_3 \psi(\boldsymbol{x}_t) \tag{7.10}$$

7.4 Facial Expression Tracking with Global and Local Fitting

Our tracking routine incorporates three components: global transformations, global deformations and local deformations. Global transformations explain rigid motion of face due to head motion. Global deformations capture nonlinear facial deformations in different expression type and in different temporal state (configuration). We further fit local deformations based the estimated global appearance model using direct local deformation estimation. If we describe the global transformation parameters by $T_{\boldsymbol{\alpha}_t}$, global shape and appearance deformation as $\boldsymbol{y}_t(\boldsymbol{a}_t, \boldsymbol{p}_t)$, and local shape fitting as $\boldsymbol{\delta}_t$, then the goal of our tracking algorithm for given image $\boldsymbol{I}_t$ is to find $\boldsymbol{\alpha}_t^*$, $\boldsymbol{p}_t^*$ and $\boldsymbol{a}_t^*$, and $\boldsymbol{\delta}_t^*$ that minimize:

$$E(\boldsymbol{\alpha}_t^*, \boldsymbol{p}_t^*, \boldsymbol{a}_t^*, \boldsymbol{\delta}_t^*) = \min_{\boldsymbol{\alpha}_t, \boldsymbol{p}_t, \boldsymbol{a}_t, \boldsymbol{\delta}_t} \left(\Upsilon(\boldsymbol{I}_t, T_{\boldsymbol{\alpha}_t} \cdot (\boldsymbol{p}_t + \boldsymbol{\delta}_t)) - \boldsymbol{a}_t\right) \tag{7.11}$$

where $\boldsymbol{a}_t^* = \boldsymbol{y}_t^*(2n+1 : N_{as})$ is an appearance vector, and $\boldsymbol{p}_t^* = \boldsymbol{y}_t^*(1 : 2n)$ is a shape vector from $\boldsymbol{y}_t^* = \boldsymbol{\mathcal{C}} \times \boldsymbol{e}^* \times \psi(\boldsymbol{x}_t^*)$ by Eq. 6.6. $\boldsymbol{e}^*$, and $\boldsymbol{x}_t^*$ are the best fitting global model estimated. $\boldsymbol{\delta}_t$ is local deformations of shape vector for given global shape and appearance deformation as described in Sec. 7.4.2.

7.4.1 Bayesian Tracking of Large Facial Deformations Using Decomposable Global Shape and Appearance Models

Given the nonlinear shape and appearance generative model, we can describe the observation of shape and appearance instance $\boldsymbol{z}_t$ by state parameters $\boldsymbol{\alpha}_t$,$\boldsymbol{x}_t$, and $\boldsymbol{e}_t$. The tracking problem is then an inference problem where at time t we need to estimate the configuration $\boldsymbol{x}_t$, facial expression type parameter $\boldsymbol{e}_t$, and global transformation $T_{\boldsymbol{\alpha_t}}$ given the observation $\boldsymbol{z}_t$. The Bayesian tracking framework enables a recursive update of the posterior $P(\boldsymbol{X}_t|\boldsymbol{Z}^t)$ over the object state $\boldsymbol{X}_t$ given all the observation $\boldsymbol{Z}^t = \boldsymbol{Z}_1, \boldsymbol{Z}_2, .., \boldsymbol{Z}_t$ up to time t:

$$P(\boldsymbol{X}_t|\boldsymbol{Z}^t) \propto P(\boldsymbol{Z}_t|\boldsymbol{X}_t) \int_{\boldsymbol{X}_{t-1}} P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) P(\boldsymbol{X}_{t-1}|\boldsymbol{Z}^{t-1})$$

In our generative model, the state $\boldsymbol{X}_t$ is $[\boldsymbol{\alpha}_t, \boldsymbol{x}_t, \boldsymbol{e}_t]$, which uniquely describes the state of the tracking object. Observation $\boldsymbol{Z}_t$ is composed of shape vector $\boldsymbol{Z}_{\boldsymbol{p}_t}$ and appearance vector $\boldsymbol{Z}_{\boldsymbol{a}t}$ from given image at time t.

The state $\boldsymbol{X}_t$ can be sub grouped into global transformation parameter $\boldsymbol{\alpha}_t$ and states for global deformations $\boldsymbol{x}_t$, and $\boldsymbol{e}_t$. The global transformation parameter is independent to the global deformation state since we can combine any shape and appearance model with any geometrical transformation to synthesize a new shape and appearance in the image space. However, they are dependent given the observation $\boldsymbol{Z}_t$. We approximate the joint posterior distribution $P(\boldsymbol{\alpha}_t, \boldsymbol{x}_t, \boldsymbol{e}_t|\boldsymbol{Z}^t) = P(\boldsymbol{\alpha}_t, \boldsymbol{y}_t|\boldsymbol{Z}^t)$ by two marginal distribution $P(\boldsymbol{\alpha}_t|\boldsymbol{y}_t^*, \boldsymbol{Z}^t)$ and $P(\boldsymbol{y}_t|\boldsymbol{\alpha}_t^*, \boldsymbol{Z}^t)$, where $\boldsymbol{\alpha}_t^*$, and $\boldsymbol{y}_t^*$ are representative values like maximum a posteriori estimations.

Observation model measures state $\boldsymbol{X}_t$ by updating the weight $\pi_t^{(i)}$ in the particle filter by measuring the observation likelihood $P(\boldsymbol{Z}_t|\boldsymbol{X}_t^{(i)}) = P(\boldsymbol{Z}_t|\boldsymbol{\alpha}_t, \boldsymbol{y}_t)$. We can estimate the likelihood by

$$P(\boldsymbol{Z}_t|\boldsymbol{\alpha}_t, \boldsymbol{y}_t) \propto \exp\left(-\frac{||\Upsilon(\boldsymbol{I}_t, T_{\boldsymbol{\alpha}_t} \cdot \boldsymbol{p}_t) - \boldsymbol{a}_t||}{\sigma}\right) \tag{7.12}$$

where $\boldsymbol{p}_t = \boldsymbol{y}_t(1:2n)$, $\boldsymbol{a}_t = \boldsymbol{y}_t(2n+1:N_{as})$, and σ is scaling factor for the measured image distance.

Particle filter for geometric transformation

We estimate global geometric transformation using particle filter based on the predicted global shape and appearance. The global shape and appearance $\boldsymbol{y}_t^{'}$ is estimated from previous estimated expression state $\boldsymbol{e}_{t-1}$, and predicted configuration $\boldsymbol{x}_t^{'}$. We assume that expression state change smoothly, and configuration change explains temporal variation given expression state. This predicted shape and appearance $\boldsymbol{y}_t^{'}$ is used as representative value $\boldsymbol{y}_t^*$. Geometric transformation state $\boldsymbol{\alpha}_t$ represents similarity transformation parameters γ_t,θ_t, and $\boldsymbol{\tau}_t$ for scaling, rotation, and translation. The marginal probability distribution represented by N_α particles $\{\boldsymbol{\alpha}_t^{(i)}, {}^{\boldsymbol{\alpha}}\pi_t^{(i)}\}_{i=1}^{N_\alpha}$. We update weights ${}^{\boldsymbol{\alpha}}\pi_t^{(i)}, i = 1, 2, \cdots, N_\alpha$ with $\boldsymbol{y}_t^{'}$ using Eq. 7.12. In the next time step, we perform important sampling with re-sampling and drifting with random walk.

Rao-Blackwellised particle filter for global deformation tracking

For the global deformation state estimation, we utilize Rao-Blackwellised particle filtering. In order to estimate global deformations using the generative model in Eq. 6.6, we need to estimate state vector $\boldsymbol{x}_t$, and $\boldsymbol{e}_t$ whose dimensions are 2, and N_e respectively. The dimension of the expression state N_e is dependent on the number of expression type which can be high dimension. When we know the configuration vector $\boldsymbol{x}_t$, we can achieve approximate solution for expression vector as explained in the followings. Original Rao-Blackwellised particle filtering for dynamic Bayesian networks [97] assumes accurate solution for the rest of part, which does not represented by particle state. We utilize the approximate solution to avoid sampling for high dimensional state density estimation, which requires huge number of particles for appropriate approximation.

Configuration $\boldsymbol{x}_t$ is embedded in 2 dimensional space with one constraints for unit circle embedding. So, the dimension of embedding is actually one-dimensional and we can represent the embedding parameter β_t as one-dimensional state vector. We represent the distribution of configuration embedding β_t by N_β particles $\{\beta_t^{(i)}, {}^{\beta}\pi_t^{(i)}\}_{i=1}^{N_\beta}$. If we represent the approximate estimation of expression vector as $\boldsymbol{e}_t^*$, we can approximate the marginal distribution

$$\begin{aligned} P(\boldsymbol{e}_t^*|\boldsymbol{y}_t) &= \sum_{\beta} P(\boldsymbol{e}_t^*|\beta_t, \boldsymbol{y}_t) P(\beta_t|\boldsymbol{y}_t) \\ &= \sum_{\beta} P(\boldsymbol{e}_t^*|\beta_t, \boldsymbol{y}_t) \sum_{i=1}^{N_\beta} {}^{\beta}\pi_t^{(i)} \delta(\beta_t^{(i)}, \beta_t) \\ &= \sum_{i=1}^{N_\beta} {}^{\beta}\pi_t^{(i)} P(\boldsymbol{e}_t^*|\beta_t^{(i)}, \boldsymbol{y}_t), \end{aligned}$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise.

We represent estimated expression vector by linear weight sum of known expression vectors. We assume that optimal expression vector can be represented as a linear combination of expression classes in the training data. i.e., we need to solve for linear regression weights κ such that $\boldsymbol{e}^{new} = \sum_{k=1}^{K_e} \kappa_k \boldsymbol{e}^k$ where each $\boldsymbol{e}^k$ is one of K_e expression classes. As the configuration is estimated already, we can obtain expression conditional class probabilities $p(\boldsymbol{e}^k|\boldsymbol{y}_t, \boldsymbol{x}_t)$ which is proportional to observation likelihood $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{e}^k)$. Such likelihood can be estimated assuming a Gaussian density centered around $\boldsymbol{\mathcal{A}} \times \boldsymbol{e}^k \times \psi(\boldsymbol{x}_t)$, i.e.,

$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{e}^k) \approx \mathcal{N}(\boldsymbol{\mathcal{A}} \times \boldsymbol{e}^k \times \psi(\boldsymbol{x}_t), \Sigma^{\boldsymbol{e}^k}).$$

Given expression class probabilities, we can set the weights to $\kappa_k = p(\boldsymbol{e}^k \mid \boldsymbol{y}_t, \boldsymbol{x}_t)$.

7.4.2 Local Facial Motion Tracking

We perform local facial motion tracking in order to estimate local deformations different from global model used for training and refine inaccurate estimation of the global transformation and the global deformation. As we approximate the joint state for the global transformation and the global deformation by sub-state distributions with representative value of the other sub-state, the actual state estimation using particle filter with limited number of particle samples shows sometimes misalignment of global transformations and inaccurate estimations of global deformations. In addition, actual motion in new sequence will variant from the global model used for learning the model even in the same person with the same expression type. Therefore, we need local facial motion tracking to refine global tracking result.

Recently, a framework to perform tracking of non-rigid object motion using TPS parameters and image gradients from a given initial frame was proposed [84]. The framework use a single initial frame as a template assuming a constant appearance during tracking after the appearance is warped back to the initial template shape. However, it does not work well in facial expression tracking when it has large shape and appearance variations. For example, a surprise expression usually makes a dramatic change of shape and appearance around mouse area, which cause the failure in fixed template based approaches. In addition, the approach presented in [84] use regular grid control points, where the tracking is affected by the background image in addition to the interested facial deformation.

We propose template-adaptive local facial motion tracking using thin-plate spline(TPS) warping. We utilize landmark points in the facial shape description as control points in TPS. The shape-normalized appearance is used as a template for local facial motion tracking. We use shape-normalized appearance as a template for local deformation estimation in facial expression tracking. The relation between warping coordinate and control points need to be computed only once as we have a unique normalized shape template across different appearance templates. The tracking result of global deformation from Our nonlinear shape and appearance model provides new appearance templates in every frame based on estimated states. The landmark shape estimated from global deformation provides initial shape for local facial motion tracking after applying global transformations to the landmark points.

Let the estimated global shape and appearance is $\boldsymbol{y}_{t0}^{g}$, its shape vector is $\boldsymbol{p}_{t0}^{g}$, and appearance vector is $\boldsymbol{a}_{t0}^{g}$, and current input image be $\boldsymbol{I}_t$, the objective of local fitting is to minimize the following error function

$$\begin{aligned} E(\delta\boldsymbol{p}_t) &= \sum \|\Upsilon(\boldsymbol{I}_t, \boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p}_t) - \boldsymbol{a}_{t0}^{g}\| \\ &= \sum_{\boldsymbol{\xi} \in \boldsymbol{p}_0} \|\boldsymbol{I}_t(W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p})) - \boldsymbol{I}_{t0}^{g}(\xi)\|^2 \end{aligned} \tag{7.13}$$

where $\boldsymbol{I}_{t0}^{g}$ is an image in normalized shape with global appearance vector $\boldsymbol{a}_{t0}^{g}$. We use shape normalized appearance as the template in local tracking, therefore, the TPS warping $W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p})$ is determined by the coordinate control points $\boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p}$ in Eq. 7.2. For the given $\boldsymbol{p}_{t0}$ from global deformation tracking, the warping function solely determined by the local deformation $\delta\boldsymbol{p}$.

Gradient descent technique is applied to find the local fitting parameter $\delta\boldsymbol{p}$ which minimize Eq. 7.13 similar to [84, 50]. Linearization is carried out by expanding $\boldsymbol{I}_t(W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p}))$ in a Taylor series about $\delta\boldsymbol{p}$,

$$\boldsymbol{I}_t(W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g} + \delta\boldsymbol{p})) = \boldsymbol{I}_t(W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g})) + \delta\boldsymbol{p}^{\mathsf{T}}\boldsymbol{M}_t + \text{h.o.t}, \tag{7.14}$$

where $\boldsymbol{M}_t = [\frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{p}_1} | \frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{p}_2} | \cdots | \frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{p}_{2n}}]$. Each term $\frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{p}_k}$ can be computed using warped image coordinate $\boldsymbol{\xi} = W(\boldsymbol{\xi}, \boldsymbol{p}_0; \boldsymbol{p}_{t0}^{g})$ by applying chain rule: $\frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{p}_k} = \frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{p}_k}$. The $\frac{\partial \boldsymbol{I}_t}{\partial \boldsymbol{\xi}}$ is the gradient of current input image $\boldsymbol{I}_t$ after TPS warping to the mean shape. The relation between warping coordinate and control points is described by Eq. 7.1 and the coefficients is computed by Eq. 7.2 are fixed and can be pre-computed as we use common mean shape in all the appearance templates. The solution for Eq. 7.13 can be computed when the higher-order terms in Eq. 7.14 is ignored:

$$\delta\boldsymbol{p} = (\boldsymbol{M}_t^{\mathsf{T}}\boldsymbol{M}_t)^{-1}\boldsymbol{M}_t^{\mathsf{T}}\delta\boldsymbol{I}_t, \tag{7.15}$$

where $\delta\boldsymbol{I}_t$ is the image difference between template appearance image and current image warped to the template shape by Eq. 7.13. By iterative update of the local shape model, we achieve better fitting of the shape to local image features. This local fitting provides better alignment of shape to the given image and better normalized appearance for the given input image.

Refinement of global state estimation based on local fitting

We update global deformation state based on estimated deformation. We estimate new expression weight with new appearance vector after local fitting. New estimated expression weight is updated by linear combination of local expression weight $\boldsymbol{\kappa}^l$ and global expression weight $\boldsymbol{\kappa}^g$, $\boldsymbol{\kappa}^{new} = (1-\varepsilon)\boldsymbol{\kappa}^g + \varepsilon\boldsymbol{\kappa}^l$. The combining parameter $\varepsilon$is depend on the reliability of local fitting. For example, for unknown subject, local fitting is less reliable and we assign small value for ε. This refined global state estimation helps the estimation of geometry transformation in the next step.

7.5 Experimental Results

To build nonlinear generative model of facial expressions and to test facial expression recognition, we selected 40 sequences of 10 female subjects ($N_s = 10$) with four emotions ($N_e = 4$: surprise, happy, angry and sad) from Cohn-Kanade AU coded facial expression database [66]. We manually marked shape landmark for every other frame for normalized dynamic shape and appearance. As the database have sequences from a neutral expression and to the peak expression, we embed each frame on the half circle with equal distance for each sequence.

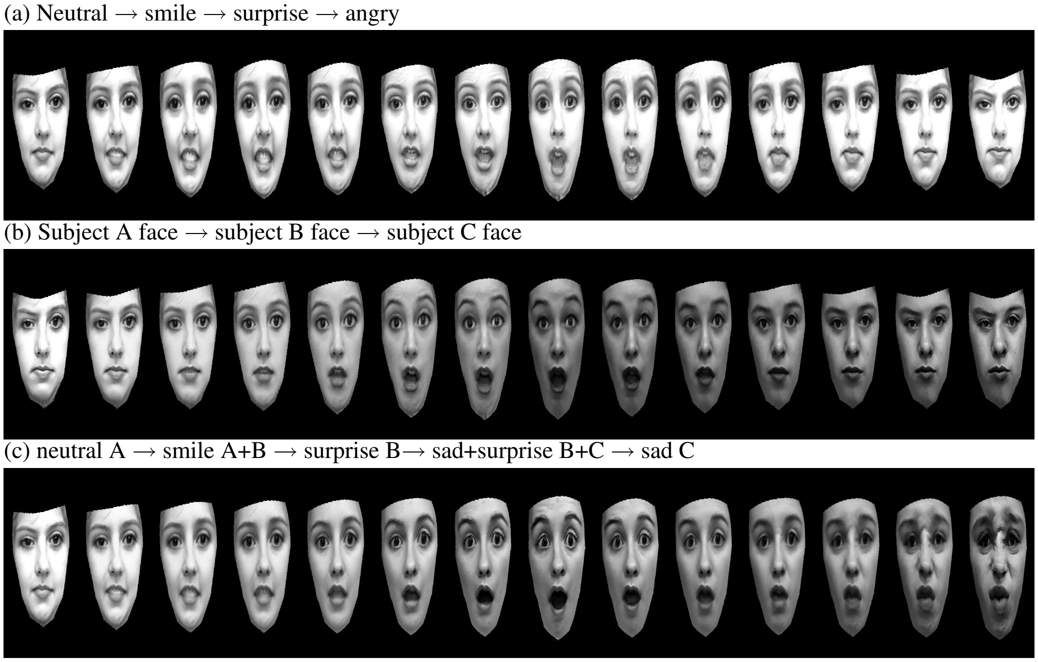
7.5.1 Facial Expression Analysis

We collected shape landmark points for every other frame in each sequence. Collected sequence data contains 5 to 17 image frames in all the sequences. The total number of frame collected with landmark points was 399 frames. The landmarks have 38 points in each image ($n = 38$). The appearance vector was represented by 288328 pixels inside the mean shape ($N_a = 288328$).

For any given sequence, we embed expression frames on the half circle as expression sequences in the database start from neutral expression and stop in the peak expression instead of returning to the neutral expression. The embedding configuration is parameterized in the range $\gamma = [0, 1]$ to cover half unit circle embedding space. So, $\gamma = 0$ means neutral configuration and $\gamma = 1$ means peak configuration.

Synthesis of facial expression with personal style and expression type variations

Our generative model can synthesize facial expression while changing the person style and expression type parameters during performing the expression. Fig. 7.4 shows synthesis examples of new facial expressions and personal faces. During synthesis of the new images, we combine control parameter t to embedding configuration γ and interpolation parameter α and β. In case of Fig. 7.4 (a), the t changed $0 \rightarrow 1 \rightarrow 0$ and new expression parameter $\boldsymbol{e}_t^{new} = (1-t)\boldsymbol{e}^{smile} + t\boldsymbol{e}^{surprise}$ and then $\boldsymbol{e}_t^{new} = (1-t)\boldsymbol{e}^{angry} + t\boldsymbol{e}^{surprise}$. As a result, the facial expression starts from neutral expression of smile and animates new expression as t changes. When $t = 1$, the expression becomes a peak expression of surprise, then the expression t changes to angry and then back to neutral expression again. In the same way, we can synthesize new faces during smile expressions as in (b). Fig. 7.4 (c) is the simultaneous


(a) Neutral → smile → surprise → angry

(b) Subject A face → subject B face → subject C face

(c) neutral A → smile A+B → surprise B→ sad+surprise B+C → sad C

Figure 7.4: Facial expression synthesis: First row: Expression transfer. Second row: Personal face transfer during smile expression. Third row: simultaneous transfer of expressions and personal faces.

MT	Non-GM				MLA			
ET	SP	HP	AG	SD	HP	AG	SD	SP
SP	**9**	0	0	1	**9**	1	0	0
HP	0	**10**	0	0	0	**10**	0	0
AG	0	1	**8**	1	0	1	**7**	2
SD	0	0	1	**9**	0	0	4	**6**

Table 7.1: Facial expression recognition with a peak expression image: Non-GM: Nonlinear Generative Model (proposed method), MLA: Multilinear Analysis in [140], MT: Applied recognition method, ET: Expression type, SP: Surprise, HP: Happy, AG: Angry,SD: Sad

control of the personal face and expression parameters. In this case, the embedding changed from $0 \rightarrow 1 \rightarrow 0.5 \rightarrow 1$. As a result, the last synthesized expression is the peak expression of the target expression instead of a neutral expression.

Facial Expression Recognition

We tested facial expression recognition performance for ten subjects with four expressions: surprise (SP), happiness (HP), angry (AG), and sadness (SD). Using collected shape normalized appearance, the performance was tested by leave-one-out method: we learn the model with nine people and tested the recognition performance with one person whose data are not used for training. To compare the performance of facial expression in [140], which applied multilinear model for AAM model of peak expression, we classified facial expressions using maximum expression weight of the last frame (peak expression frame) in every sequence even though we estimated expression class weight for every frame as described in Sec. 7.3.1. Table 7.1 shows recognition results.

The average recognition rate in our method is $90\%(\frac{36}{40})$, which is better than multilinear model$\big(80\%(\frac{32}{40})\big)$, where facial expressions for unknown person are recognized based on cosine distance of estimated expression vector using one of closest person subspace with the same shape normalized appearance data. Our model has a better style and expression decomposition model as our model use all the image sequences with different number of frames for training, which is impossible in multilinear analysis in [140] as it requires aligned the same number of frames for training multilinear analysis.

7.5.2 Facial Expression Tracking

We describe global facial motion deformations by linear combinations of expression classes. Fig. 7.5 shows tracking a smile expression sequence with template adaptive local fitting. At each frame, global facial expression tracking estimates expression weights (c) and configurations after global transformation estimation. The best fitting shape appearance parameter provides the shape normalized appearance template (a) and facial shape tracking after global shape deformation (b). Results of local fitting in Fig. 7.5 (e) shows better fitting of shape deformation estimation to the input image and better estimation of facial expression type (f). Facial expression weight only using global deformation estimation has inaccurate similar weights in 'surprise' and 'happiness (smile)'. Whereas, after local fitting, the estimated expression type distinguishes two expressions and shows high weights for happy expression correctly. After updating estimated expression type by a linear combination of the global deformation and the local deformation as described in Sec. 7.4.2, we achieve better estimation of global facial motion tracking (i).

Tracking large facial deformations

We compare tracking results with single template and adaptive template in large facial deformations. Fig. 7.6 (b) is facial motion tracking result based on single frame. It shows good tracking of facial deformations in small deformations. However, it fails to track large facial deformations around mouse area. Fig. 7.6 (c) shows tracking result when we use global transformation result as intimal state of local facial motion tracking in each frame. As global deformation model provides updated appearance template in addition to initial shape for tracking, it can cover large facial deformations.

7.6 Summary

We presented a new approach for facial expression recognition and synthesis. The model utilized nonlinear warping of appearance for shape normalized appearance model and kernel mapping to model nonlinearity of appearance in facial expressions. The dynamics of facial expressions are also modeled using low dimensional manifold embedding of the expression

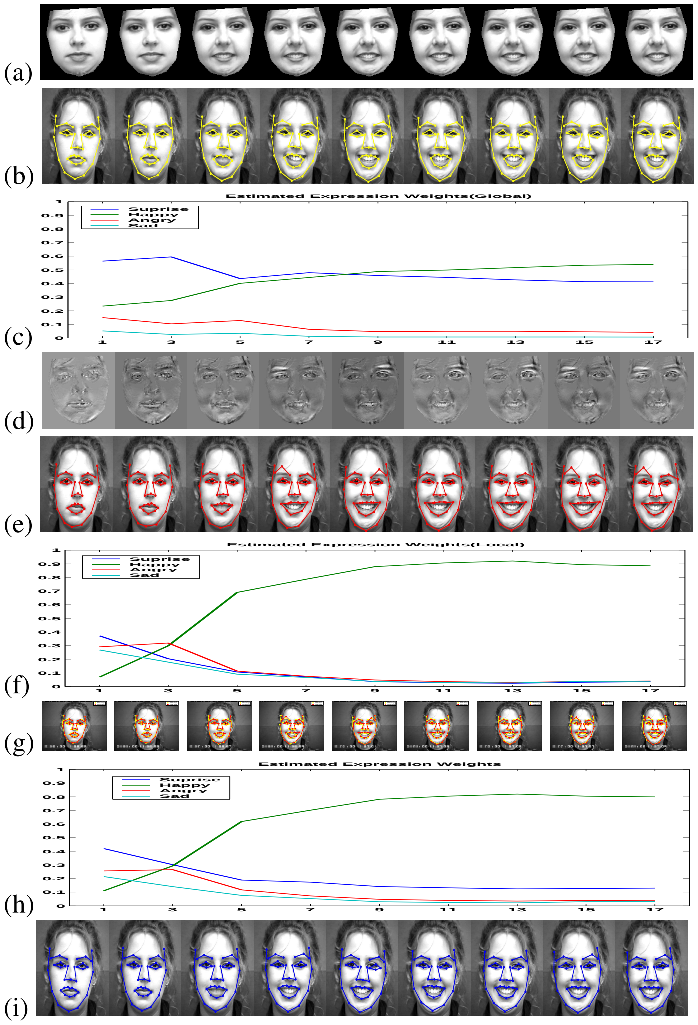


Figure 7.5: Facial expression tracking with global and local fitting: (a) Best fitting global appearance in normalized shape. (b) Global shape tracking facial motion. (c) Expression weights in global facial motion estimation. (d) Image error in the local fitting. (e) Local tracking facial motion with adaptive template provided by global appearance model. (f) Expression weights in local facial motion estimation. (g) Comparison of tracking result: yellow-global fitting, red-local fitting. (h) Update of estimated expression weights by combination of local and global expression estimation. (i) Best fitting global model using updated expression state.

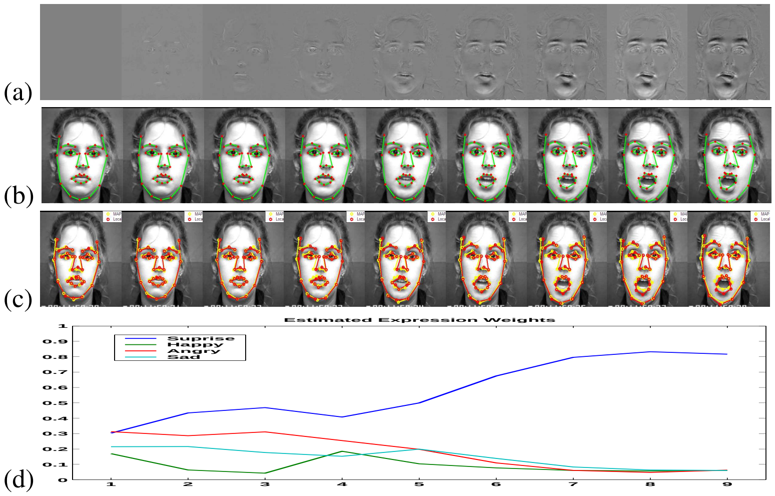


Figure 7.6: Tracking surprise expression : (a) Error image based on a template after local fitting. (b) Tracking result by direct local fitting with initial frame as a template. (c) Tracking result with adaptive template by global shape and appearance model: yellow-global fitting, red-local fitting. (d) Global estimated expression weights.

configuration. The model shows better performance in facial expression recognition in addition to accurate synthesis of facial expressions with simultaneous geometry and expression control. Using the proposed generative model, which has a low dimensional representation of dynamics with preserving nonlinearity and dynamics, we presented a framework for facial motion tracking in large facial deformations. Global tracking provides template adapting to appearance change in large deformation. Local fitting with templates from global deformation enables accurate fitting from coarse global shape according to local image appearance.

Chapter 8

Applications

8.1 Scalable View-invariant Gait Recognition

Human identification using gait is a challenging computer vision task due to the dynamic motion of gait and the existence of various sources of variations such as viewpoint, walking speed, walking surface, clothing, etc. In this section we present gait recognition system based on temporal normalization and style analysis. We develop a generative model by embedding gait sequences into unit circles and learning nonlinear mappings which facilitate synthesis of temporally-aligned gait sequences. The bilinear analysis of temporally-aligned gait sequences decompose gait into time-invariant gait style and time-dependent gait content factors. We extend the bilinear gait model into tensor gait model, multilinear decomposition of gait sequences, for view-invariant gait style representation. Given walking sequences captured from multiple views of multiple people, we fit a multilinear model using higher-order singular value decomposition which decomposes view factors in addition to the body configuration and gait style factors. Gait style is a view-invariant, time-invariant, and speed-invariant gait signature that can then be used in recognition. In the recognition phase, a new walking cycle of unknown person in unknown view is automatically aligned to the learned model after cycle detection and then iterative procedure is used to solve for both the gait style parameter and the view. The estimated gait style parameters are used as feature vector for gait recognition. We also show that the recognition can be generalized to new environment conditions by adapting the gait content factor to reflect new observation condition and therefore obtain more accurate gait-styles estimation and recognition. The proposed framework allows scalability to add a new person to already learned model even if a single cycle of a single view is available. We present experimental result using CMU Mobo gait database and USF gait challenging database.

8.1.1 Manifold Embedding and Temporal Normalization of Gait

We explore gait embedding in low dimensional manifold space and achieve temporal normalization based on manifold embedding and resampling.

Manifold Embedding for Gait

In order to achieve a recognition task with the existence of twists in the nonlinear embedded manifolds, we need to use a standardized embedding that approximates the original manifold. These variations pose a challenge if we would like to use motion manifolds as constraints for the synthesis with temporal normalization. But, conceptually all these manifolds are the same. They are all topologically equivalent, i.e., homeomorphic to each other and we can establish a bijection between any pair of them. They are all also homeomorphic to the gait manifold in a kinematic 3D body configuration space. Therefore, we embed each half gait cycle temporally on a unit circle, i.e. a one dimensional manifold embedded in a two dimensional space. Input silhouettes corresponding to each half walking cycle are embedded on an equally spaced points along a unit circle for side view gait analysis. For view variant sequence, we use full one cycle for the embedding.

Temporal Normalization and Re-Sampling

Given manifold embedding, we need to synthesize new silhouettes at standard time instances during the cycle to be used for recognition. We define *N-synthesized gait poses* as a collection of N synthesized silhouettes at N equally spaced time instances during a cycle which indicates how the silhouette shape will look like at these N standard intermediate points. These synthesized gait poses achieve temporal normalization from different number of frames, or different walking speed, in each cycle.

In order to obtain such synthesized gait poses, we learn a nonlinear mapping function from the manifold embedded on a unit circle into the input silhouettes. Learning nonlinear mapping is necessary since the manifold is embedded nonlinearly and arbitrary into a unit circle. We use generalized radial basis function (GRBF) [103] to learn such mapping as a collection of interpolation functions.

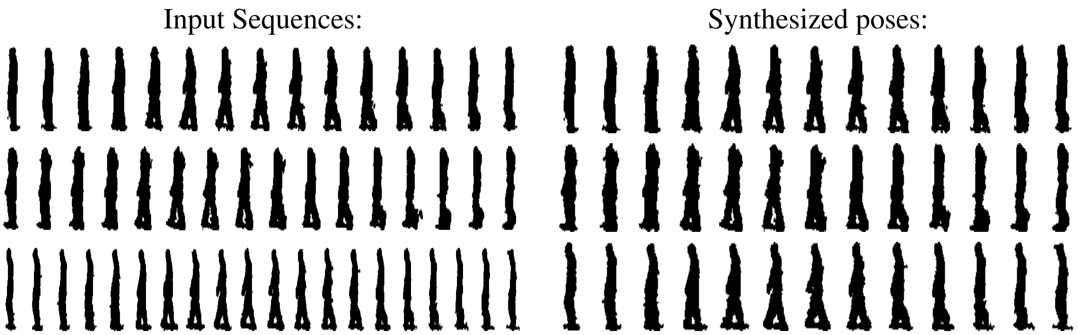


Figure 8.1: Original gait image sequences and their normalized gait poses

Let M equally spaced centers along a unit circle be $\{t_j \in \mathbb{R}^e, j = 1, \cdots, M\}$ and given a set input images $\boldsymbol{Y} = \{\boldsymbol{y}_i, i = 1, \cdots, M\}$ and let their corresponding embedding along the unit circle be $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, \cdots, M\}$, we can learn interpolants in the form

$$f^k(\boldsymbol{x}) = p^k(\boldsymbol{x}) + \sum_{i=1}^{N} w_i^k \phi(|\boldsymbol{x} - \boldsymbol{x}_i|), \tag{8.1}$$

that satisfies the interpolation condition $\boldsymbol{y}_i^k = f^k(\boldsymbol{x}_i)$ where $\boldsymbol{y}_i^k$ is the k-th pixel of input silhouette $\boldsymbol{y}_i$, $\phi(\cdot)$ is a real valued basis function, w_i^k are real coefficients, $p^k(\cdot)$ is a linear polynomial, and $|\cdot|$ is the Euclidean distance, L^2-norm. The mapping coefficients can be obtained by solving a linear system of equations as shown in Sec. 3.3. Such mapping can be written in the form of a generative model as

$$f(\boldsymbol{x}) = \boldsymbol{B} \cdot \psi(\boldsymbol{x}) \tag{8.2}$$

that nonlinearly maps any point $\boldsymbol{x}$ from the two dimensional embedding space into the input space and therefore can be used to synthesize N intermediate silhouettes at N standard time instances equally spaced along the unit circle. Therefore, Re-sampled gait from the embedding space enables us to find temporally well aligned gait poses invariant to different walking speed and frame rate using equally spaced N embedding points. In Fig. 8.1, the left three rows show original image sequences for three different people and the right three rows show N-normalized gait poses synthesized using the learned models from each input sequence.

Bilinear Model for Gait

It is well known in psychology that human perceptual systems naturally separate the content and style factor of their observation in identifying a familiar face or gait seen under unfamiliar

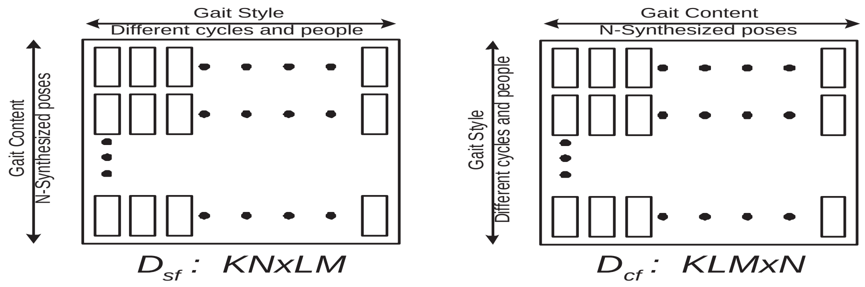


Figure 8.2: Style format and content format

viewing conditions. In the context of gait we aim to separate two orthogonal factors: *gait style:* time-invariant personalized style of the gait which can be used for identification; and *gait content:* time-dependent factor representing different body poses during the gait cycle. Gait content is also dependent on other conditions such as viewpoint, shoe, ground, etc. An input silhouette can be represented by a bilinear model as

$$\boldsymbol{I}_{sc} = \sum_{i=1}^{N} \sum_{j=1}^{J} w_{ij} \boldsymbol{c}_i \boldsymbol{s}_j \tag{8.3}$$

using gait style vectors $\boldsymbol{s}$ and gait content vectors $\boldsymbol{c}$ and basis images w_{ij}, i.e., the model linearly combines basis images w_{ij} using the style coefficients $\boldsymbol{s}_j$ and content coefficient $\boldsymbol{c}_i$. The gait content vector varies with time through the walking cycle to generate the various body poses observed through the walking given the time-invariant gait style vector that characterizes the walker.

Given a training data with different people and multiple gait cycles per person which might manifest different conditions, the objective is to fit a model in the form of Eq. 8.3. The first step towards this is to warp the time domain of different cycles to establish correspondences in time between different cycles. This is done by embedding each cycle on a unit circle and therefore synthesize intermediate poses at standard time instances to represent each cycle in the training set.

Given L gait cycles for each of M different people in the training data where each gait cycle is represented as N synthesized time-aligned poses at N standardized time instances, where each image is represented as a K dimensional vector, we aim to fit a symmetric bilinear model in the form of Eq. 8.3.

We arrange the synthesized gait image sequence into two forms: one is *style format*, $\boldsymbol{D}_{sf}$, the other is *content format*, $\boldsymbol{D}_{cf}$, as shown in Fig. 8.2. In style format, we have LM columns where each column contains N synthesized gait pose images as one gait cycle vector and the column vector size is KN. In content format, we have N columns where each column represents images of the same synthesized gait pose from all of the different gait cycles and different people gait sequences, i.e., each column is of KLM dimension.

Given such arrangement, the objective is to decompose the style and content vectors, i.e., to decompose the matrix $\boldsymbol{D}_{sf}$ as

$$\boldsymbol{D}_{sf} = \boldsymbol{C}\boldsymbol{W}_{cs}\boldsymbol{S} \tag{8.4}$$

or similarly $\boldsymbol{D}_{cf} = \boldsymbol{S}\boldsymbol{W}_{sc}\boldsymbol{C}$. Such model is called a symmetric bilinear model and it is necessary in order to adapt the gait styles to new gait contents given new situations as will be discussed later. In order to achieve such decomposition, asymmetric bilinear model is used to decompose the data to separate gait style vectors $\boldsymbol{S}$ given content-dependent mapping $\boldsymbol{T}_c$ and to separate gait content vectors $\boldsymbol{C}$ given style-dependent mapping $\boldsymbol{T}_s$ as

$$\boldsymbol{D}_{sf} = \boldsymbol{T}_c\boldsymbol{S} \tag{8.5}$$

$$\boldsymbol{D}_{cf} = \boldsymbol{T}_s\boldsymbol{C} \tag{8.6}$$

to minimize the reconstruction error, i.e., to minimize $E = ||\boldsymbol{D}_{sf} - \boldsymbol{T}_c\boldsymbol{S}||^2$ and similarly for $\boldsymbol{D}_{cf}$. Such decomposition can be achieved by singular value decomposition (SVD) as was shown in [128]. Given SVD for $\boldsymbol{D}_{sf}$ as $\boldsymbol{D}_{sf} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, least square optimal solution is $\boldsymbol{S} = \boldsymbol{V}^T$ and $\boldsymbol{T}_c = \boldsymbol{U}\boldsymbol{D}$. Similarly we can achieve the decomposition in Eq. 8.6. We can use J-dimensional approximation by choosing first J largest diagonal terms in D and setting the rest to zeros. Given an asymmetric model fitted in the form of Eq. 8.5 and 8.6, symmetric model can be fitted iteratively. However, the bilinear model will not works well when there are multiple variant factors in addition to the gait style.

8.1.2 Multilinear Model for Gait Analysis: TensorGait

Given different people walking sequences from different views, we detect gait cycles using gait cycle detection algorithm. After cycle detection for every person, each cycle is used to learn the generative model described by equation 8.2 and re-sampled with the same number of temporally aligned poses. Therefore, the training data consists of N_s gait cycles, each captured from N_v different views, and each consists of N_p silhouette images representing aligned body poses. Each silhouette image is represented as a d dimensional vector using the representation described in section 8.1.1. The whole collection of aligned cycles for all different people and views is arranged into order four tensor (4-way array) $\boldsymbol{\mathcal{D}}$ with dimensionality $N_s \times N_v \times N_p \times d$.

The data tensor $\boldsymbol{\mathcal{D}}$ can be decomposed to parameterize orthogonal style, view, and pose factors using higher-order singular value decomposition (HOSVD). Higher-order singular value decomposition (HOSVD) is a generalization of SVD for multilinear model analysis by [72, 138, 137]. Multilinear model is a generalization of linear model (one-factor models) and bilinear model (two-factor models) [128] into higher-order tensor decomposition (multi-factor models). The data tensor $\boldsymbol{\mathcal{D}}$ is decomposed to establish forth-order tensor using HOSVD which yields the decomposition

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{Z}} \times_{\mathbf{1}} \boldsymbol{S} \times_{\mathbf{2}} \boldsymbol{V} \times_{\mathbf{3}} \boldsymbol{P} \times_{\mathbf{4}} \boldsymbol{M}, \tag{8.7}$$

where $\boldsymbol{S}$, $\boldsymbol{V}$, $\boldsymbol{P}$, and $\boldsymbol{M}$ are orthogonal matrices with dimensionality $N_s \times N_s$, $N_v \times N_v$, $N_p \times N_p$, $d \times d$ corresponding to style, view, pose, and image orthogonal bases respectively. $\boldsymbol{\mathcal{Z}}$ is a core tensor with the same dimensionality as the data tensor $\boldsymbol{\mathcal{D}}$ which represents the interaction of the gait style, view, pose, and image pixel subspaces.

The orthogonal $N_s \times N_s$ matrix $\boldsymbol{S}$ spans the space of gait style parameters. In the style basis matrix $\boldsymbol{S} = [\boldsymbol{s}^1 \boldsymbol{s}^2 \cdots \boldsymbol{s}^s]^T$, each vector $\boldsymbol{s}^i$ represents a style parameter of person i as an N_s dimensional vector. This parameterization of the gait style independent of the view and body configuration is the basic feature we use in the recognition. Fig. 8.3 shows an example of the decomposition of gait style. We use 4 people from CMU-Mobogait data set with 6 cycles each from 4 different views to fit the model. As apparent in the figure, gait style parameters estimated from the different cycles of each person are clustered together in the style space.

Equation 8.7 can be rewritten as a generative model to synthesize gait cycles given any style vector $\boldsymbol{s}$ and view vector $\boldsymbol{v}$. This can be achieved by defining a new core tensor $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{Z}} \times_3 \boldsymbol{\mathcal{P}} \times_4 \boldsymbol{\mathcal{M}}$. Therefore, gait cycle images can be synthesized as $\boldsymbol{D}^{\boldsymbol{sv}}$ where

$$\boldsymbol{D}^{\boldsymbol{sv}} = \boldsymbol{\mathcal{B}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{v} \tag{8.8}$$

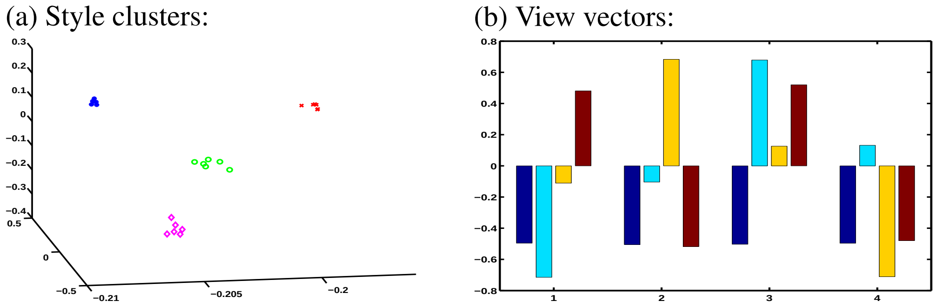

(a) Style clusters:
(b) View vectors:

Figure 8.3: Tensor analysis: 4 people with 6 cycles each from 4 different views. (a) First three style parameters for 6 gait cycles of each person. Each person's style shows good clustering within the person and good separation between different persons. (b)Four different view vectors, which are orthogonal to each others.

8.1.3 Gait Recognition Algorithm

For gait recognition, we need to estimate style factors. We presents how to estimate style from bilinear gait model and multilinear gait model. For the bilinear model, we used support vector machine in addition to nearest neighborhood classifier. The recognition algorithm for bilinear model using style vector can be applicable to multilinear gait model. In order to achieve accurate style estimation in new environment, we utilized symmetric gait style model with adaptation of content vectors in new environment. We use tensor gait model and estimate style from know and unknown view in the view variant data set.

Gait Recognition using Asymmetric Bilinear Model

Given a new probe sequence and given a learned model from the training data, the objective is to identify the person in the probe sequence, i.e., to recognize the gait style. Each probe sequence is first segmented into half cycles and each half cycle is embedded into a unit circle. Then, nonlinear mapping is used to synthesize time-aligned poses to be used in recognition. Given N-synthesized silhouettes for each cycle of the probe sequence, the data is arranged into a column vectors I^k_{probe} of dimensionality KN where K is image size (height × width) for each probe cycle k. Given the asymmetric model learned from the training data as in Eq. 8.5, we can solve for style vectors s^k_{probe} using the pseudo-inverse for the content-dependent mapping T_c, i.e.,

$$s^k_{probe} = T_c^{-1} I^k_{probe} = D^{-1} U^T I^k_{probe} \tag{8.9}$$

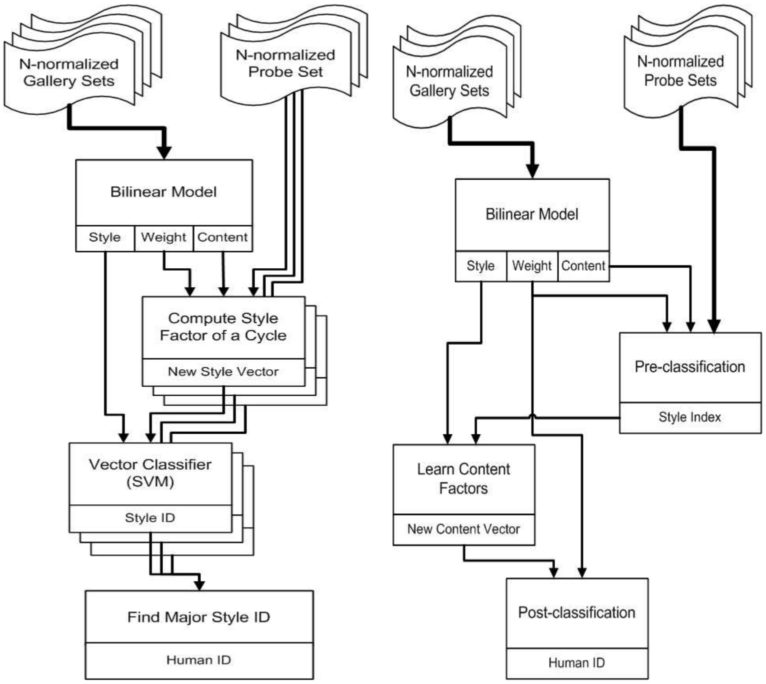


Figure 8.4: Recognition algorithm

The resulting probe style vectors, and the style vectors learned from the training data are the basic features that can be used in the recognition. Each gait style vector is a point in a J-dimensional feature space, and general classifiers can be used for classification purpose. We used both k-NN classifier and SVM classifier to classify any new probe-style vector to one of the learned people classes. The recognition procedure is shown in Fig. 8.4.

Notice that such recognition procedure uses only the asymmetric model in Eq. 8.5. Why then we need a symmetric model? The answer is that we need symmetric model to adapt the model to new environment and situations as will be discussed next.

Gait Recognition Using Symmetric Bilinear Models: Adapting Gait Style to New Situations

We expect gait style factor to be invariant to different situations such as view point, shoe, ground, etc. How can that be achieved if we do not see all these different situations in the training data? Given a learned model using data collected under certain situations, how can we use such model for recognition under different situations? New situation means new gait content or simply means new content-dependent mapping. Given a symmetric model in the

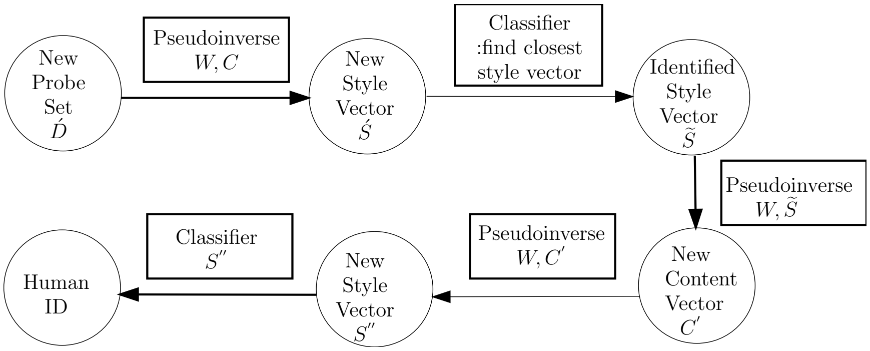


Figure 8.5: Classifier with adaptation of content vectors

form of Eq. 8.4, if we can adapt the content vector C to a new content vector $C^{'}$ for the new situation we can then solve for the style vectors under the new situation. In other words we need to extrapolate gait styles given new situations.

Given a probe data set captured under new condition, we show here how to recognize the people in the probe set by adapting the model to the new condition. The only assumption we make is that all the people in the probe set are part of the gallery set used for the training. This is necessary anyway to be able identify the people in the probe set. If we know the correspondences between people in the probe set and the people in the training set, then, we can obviously solve for the new content vectors $C^{'}$ using the inverse of the style-dependent mapping T_s. Unfortunately, we do not know such correspondences since we do not know people class for the probe set.

In order to simultaneously identify people and adapt to new situations we use the following procedure. First, the probe data set is arranged in style-format and content-format, $D_{sf}^{probe}, D_{cf}^{probe}$ as was shown in Section 8.1.1 after detecting cycles, embedding, and gait synthesis as was shown in Section 8.1.1. We can solve for approximate style vectors $\boldsymbol{S}^{'}$ for the probe set by pseudo-inverse using original $\boldsymbol{W}, \boldsymbol{C}$ as

$$\boldsymbol{S}^{'} = [(\boldsymbol{W}^{\nu}\boldsymbol{C})^{\nu}]^{-1}\boldsymbol{D}_{sf}^{probe} \tag{8.10}$$

where ν is matrix vector-transpose operation as defined in [128]. Given the recovered styles $\boldsymbol{S}^{'}$ we can classify each cycle in the probe set to identify corresponding person from the training data. We call this step pre-classification. Given the pre-classification result we can recover the original style vectors $\tilde{\boldsymbol{S}}$ by finding closest style vector in the original style vectors for each probe cycle. Now, we can adapt content vectors to a new situation by solving for the new content vectors $\boldsymbol{C}^{'}$ using the recovered style vector as

$$\boldsymbol{C}^{'} = [(\boldsymbol{W}\tilde{\boldsymbol{S}})^{\nu}]^{-1}\boldsymbol{D}_{cf}^{probe} \tag{8.11}$$

This adapted content vectors $\boldsymbol{C}^{'}$ are expected to represent new environment better than original content vectors $\boldsymbol{C}$. Finally, we can use the adapted content vectors $\boldsymbol{C}^{'}$ to obtain new style vectors $\boldsymbol{S}^{''}$ in the same way as in Eq. 8.10 which is then can be used for final classification in the new environment. The procedure is shown in Fig. 8.5. This iterative procedure can be repeated to obtain better results.

Gait Recognition using Multilinear Analysis: Style Estimation for Unknown View

Given images $y_1, \cdots, y_k$ representing a full gait cycle from unknown view with k frames, estimation of gait style is required for person identification. First, the sequence is used to learn a generative model in the form of Eq. 8.2 and then the model is used to re-sample p gait images, $\boldsymbol{i}_1\boldsymbol{i}_2\cdots\boldsymbol{i}_p$, which are aligned with gait poses used in multilinear analysis. By stacking the gait images into a matrix $\boldsymbol{D} = [\boldsymbol{i}_1\boldsymbol{i}_2\cdots\boldsymbol{i}_p]$, the estimation of style and view can be formulated as solving for $\boldsymbol{s}$ and $\boldsymbol{v}$ that minimize error

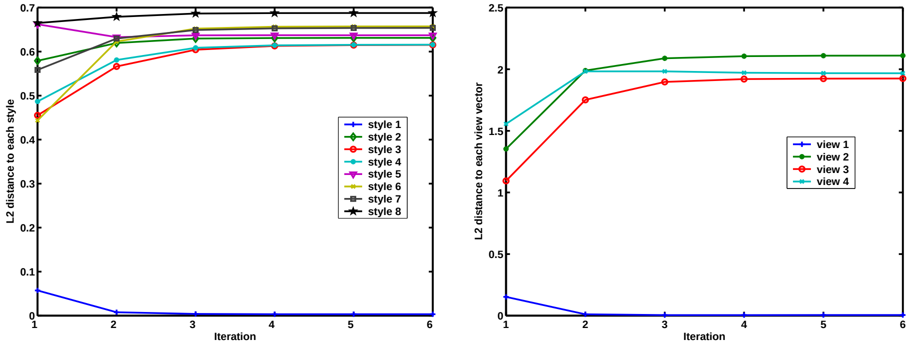
$$E(\boldsymbol{s},\boldsymbol{v}) = ||\boldsymbol{D} - \boldsymbol{\mathcal{B}} \times_1 \boldsymbol{s} \times_2 \boldsymbol{v}||, \tag{8.12}$$

where $\boldsymbol{D}$ is $d \times N_p$ matrix. If the view vector $\boldsymbol{v}$ is known, we can obtain closed form solution for $\boldsymbol{s}$. This can be done by evaluating the product $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{B}} \times \boldsymbol{v}$ and unfolding the tensor $\boldsymbol{\mathcal{H}}$ into a matrix by style-mode, i.e., $\boldsymbol{H_{(1)}} = unfolding(\boldsymbol{\mathcal{H}}, 1)$, where $unfolding(\cdot)$ is matrix unfolding operations defined in Appendix A. The dimensions of $\boldsymbol{H_{(1)}}$ is $N_s \times (N_v \times N_p \times d)$. Solution for $\boldsymbol{s}$ can be obtained in closed form by solving the linear system $\boldsymbol{D} = \boldsymbol{H_{(1)}}^T\boldsymbol{s}$. Therefore estimation of $\boldsymbol{s}$ can be obtained by

$$\boldsymbol{s} = \left(\boldsymbol{H_{(1)}}^T\right)^+ \boldsymbol{D} \tag{8.13}$$

where $+$ is matrix pseudo-inverse operation using singular value decomposition (SVD). Similarly, we can analytically solve for $\boldsymbol{v}$ if the style vector $\boldsymbol{s}$ is known by forming a tensor $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{B}} \times_1 \boldsymbol{v}$ and forming its view-mode unfolding $\boldsymbol{G_{(2)}}$. Therefore, we can obtain the view as

$$\boldsymbol{v} = \left(\boldsymbol{G_{(2)}}^T\right)^+ \boldsymbol{D} \tag{8.14}$$


(a) L_2 distance to each style vector in iteration:

(b) L_2 distance to each view vector in iteration:

Figure 8.6: Measurement of distance to style and view

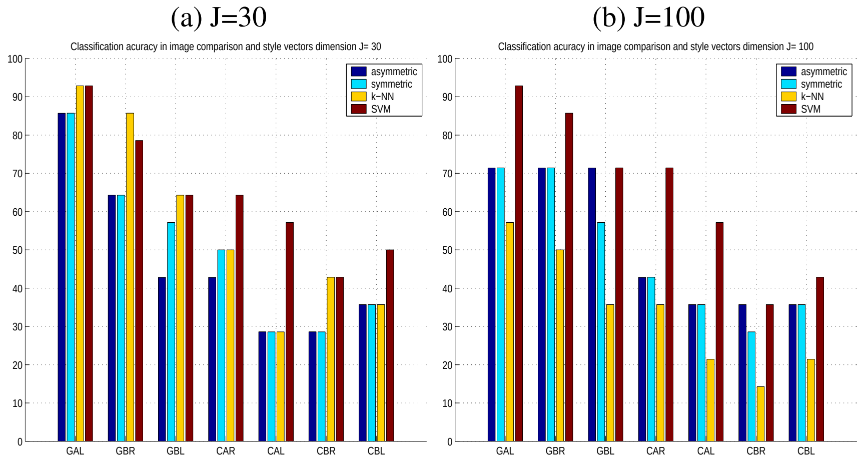
Iterative estimation of $\boldsymbol{s}$ and $\boldsymbol{v}$ using (8.13) and (8.14) leads to local minima for the error in (8.12). We can start initial style estimation by mean style $\boldsymbol{s} = (\sum_{i=1}^{N_s} s^i)/N_s$. Fig. 8.6 shows an example of the iterative estimation of view and gait style parameters. In this experiment we used 8 people with 4 different views from the Mobogait dataset to learn the model. The figure shows the change in the Euclidean distance to each mean style vector and mean view vector with the iterations. In this figure, a side view cycle for the first person was used for testing. It shows convergence to the correct style and view from the first iteration.

Given the estimated gait style vector $\boldsymbol{s}$, and different people's gait style vectors learned in the training, the recognition is a typical pattern classification problem. We used two simple classification approaches: Nearest Neighbor and Nearest class mean in multilinear gait model. Experimental results show good recognition results even this simple classifier using iteratively estimated style vector. More sophisticated classification methods can be applicable to achieve even better results.

The proposed framework based on bilinear and multilinear analysis and gait recognition based on estimated gait style vector can easily scale to include new people. Given a new person, theoretically, only one cycle from a single view is required to be able to solve for the person style parameter which can then be added to the trained database.

8.1.4 Experimental Result

Experimental results using CMU Mobo gait database and NIST-USF database [101] are reported in this paper. We use NIST-USF Gait database [101] to test our bilinear approach with


(a) J=30

(b) J=100

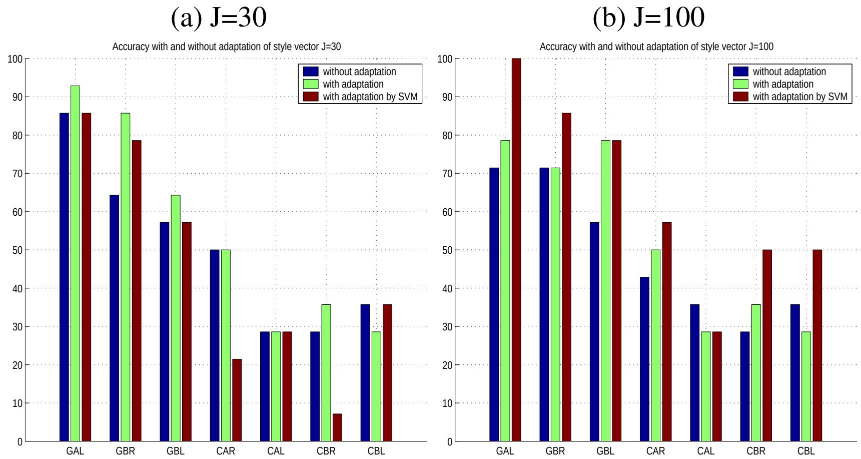
Figure 8.7: Recognition based on recovered gait style vectors.

adaptation to new environment. We use computed silhouettes for the May-Nov-2001 data. This data set has probe sets with variation of viewpoint, footwear, walking surface and with/without briefcase. We tested for the variation of viewpoint, footwear, and walking surface and compared with baseline result [112]. For the gait recognition in multiple view, we tested with CMU Mobo gait database.

Bilinear Model Test Using USF database

NIST-USF Gait database [101] is used to learn and evaluate our algorithm. We used computed silhouettes for the May-Nov-2001 data. We arbitrary select 14 peoples for preliminary evaluation from grass surface, shoes type A, and right camera sequences as a gallery set (GAR) and tested seven different conditions by variation of viewpoint (L), footwear (B), and walking surface (C). Original image size is 128×88 and were resized 64×44, i.e., each input vector size is $64 \times 44 = 2816$. The number of gait poses in synthesized gait is 13 and vector size for one gait cycle is $36608 = 2816 \times 13$. The number of style vectors are $112 = 8$(cycle per people)$\times 14$ (people). The dimension of the data set D_{sf} is 36608×112.

We evaluated identification accuracy with four different approaches: 1) An asymmetric model with nearest neighbor classifier on the recovered gait styles. 2) A symmetric model with nearest neighbor classifier. 3) A symmetric model with k-nearest neighbor classifier (k-NN). 4) A symmetric model with support vector machine classifier (SVM). For identification using gait, we need to determine people based on sequences which might be composed of several


(a) J=30

(b) J=100

Figure 8.8: Recognition with/without adaptation to new content vectors

cycles. So the classifiers we used identify people from each available input cycle, and boost the result of multiple cycles by selecting the majority of individual classification results. Figure 8.7 show classification rates using the different classifiers.

The adaptation of the gait style to new environment helps classification in variant situation. Figure 8.8 show comparison of human identification accuracy of a symmetric model without adaptation of gait content vector and with adaptation of gait content vector using nearest neighbor as well as using SVM classifier. In most of the cases, improvement can be noticed using adaptation of content vector to new situation. In the cases where pre-classification results are above 50%, the final gait recognition results show improvement because good pre-classifications make it possible to estimate the original style vectors well and, therefore, the new situation content vector can be recovered which leads to improvements in the final style classification results.

View Invariant and Scalable Gait Recognition

We demonstrate proposed view invariant and scalable gait recognition on two databases: one is CMU mobo database and the other is USF-NIST gait database. In the preprocessing step, we applied median filter to remove noisy holes and spots. Bounding boxes which cover each person silhouettes were found and normalized to fixed size. Each silhouette shape is represented by a signed-distance function.

Experiment 1: Recognition of gait in different speeds and views: In this experiment we

View class	slow walking sequences	fast walking sequences	Collins[27] (fast walking)
1(profile)	100%	88.9%	76%
2(front-right)	100%	88.9%	N/A
3(front)	100%	92.6%	100%
4(front-left)	100%	88.9%	N/A
Average	100%	90.0%	88%

Table 8.1: Gait recognition in different view and speed (CMU Data)

used CMU Mobo database, which has slow and fast walking sequences on a treadmill with six different views [48], to test gait recognition in different speeds and views. We chose a subset of 18 subjects which provided silhouettes for all different views and allowed finding proper bounding box for the subjects. Four different views (profile view, front-right view, front view, front-left view) were selected for multilinear gait analysis. Three cycles of slow walk for each person are used to learn the multilinear model parameters. In summary, the training data contains 18 people, 3 cycles each, from 4 views. Each person style is represented by the mean of the three style vectors obtained from three training cycles.

For evaluation we used three different slow-walk cycles and three fast-walk cycles for each of the 18 people with 4 views each. Overall there are 216 slow-walk evaluation cycles and 216 fast-walk evaluation cycles. For each evaluation cycle we estimate the view and the style of parameters of gait as described in Section 8.1.3. Finally, people are identified by finding closest style class mean. Table 8.1 shows the experiment result. For the slow-walk we achieve 100 % correct recognitions for all the views. For the fast-walk, we achieve around 90 % accuracy in average. The results shows fairly consistent recognition for all the different views. In both cases we achieve 100% view identification. Even though we perform recognition for each cycle without knowing the view label, our results show better identification than template matching of key frames by Collins [27], shown in the forth column, which is tested for profile and front view separately using whole sequences.

Experiment 2: Generalization and Scalability across Different Views: We evaluate the scalability of the proposed framework, i.e., given a learned model, can we extend it to recognize a new person from different view given that only one gait cycle from a single view is available for that person for training?

View class	V1:slow	V2:slow	V3:slow	V4: slow	V1:fast	V2:fast	V3:fast	V4:fast
V1(profile)	96.3%	72.2%	53.7%	75.9%	53.7%	55.6%	40.7%	55.6%
V2(front-right)	72.2%	88.9%	59.3%	66.7%	53.7%	64.8%	48.2%	63.0%
V3(front)	51.9%	66.7%	90.9%	57.4%	50.0%	59.3%	92.6%	53.7%
V4(front-left)	59.3%	75.9%	70.7%	98.1%	46.3%	46.3%	55.6%	63.0%
Average(all)	69.9%	75.9%	68.7%	87.5%	50.9%	56.5%	59.3%	58.8%

Table 8.2: Gait recognition across different views(CMU Data)

Difference	Probe Set	Baseline	Nearest Mean	Nearest Neighbor	Kale [65]
View	GAL	73%	86%	96 %	89 %
Shoe	GBR	78%	82%	86 %	88 %
Shoe, view	GBL	48%	68%	75 %	68 %
Surface	CAR	32%	32%	43 %	35 %
Surface, shoe	CBR	22%	43%	43 %	28 %
surface, view	CAL	17%	25%	21 %	15 %
Surface, shoe, view	CBL	17%	25%	29 %	21 %

Table 8.3: Comparison of Recognition with Baseline (USF Data)

To evaluate this, we performed a new experiment by learning the model with a subset of subjects. Among 18 subjects, we learned the model using only eight subjects' slow walk sequences from 4 views. For the rest 10 subjects, only a single cycle data of slow walk from one view was given. We used this single view cycle to estimate gait style parameters. All the estimated style parameters are used as a database for recognition. The recognition is then evaluated using a test set consisting of 3 different slow-walk cycles and 3 fast-walk cycles from 4 views for all the 18 people.

Table 8.2 shows recognition results across different views. We repeated the experiment by varying the view used in training for the 10 people with each single view cycle. Results show general identification capability to unknown views using style learned from a specific view. This clearly shows that the gait-style parameter is invariant to different view point. The identification performance varies across different views and the view used for training shows better performance on trained view class than others. Others, which do not learned style at all for the views,still, shows potentials for gait recognition. The performance can be improved by using multiple cycles in the style estimation for given views.

Experiment 3: Recognition of Gait with Continuous Variation of Views (USF dataset): In

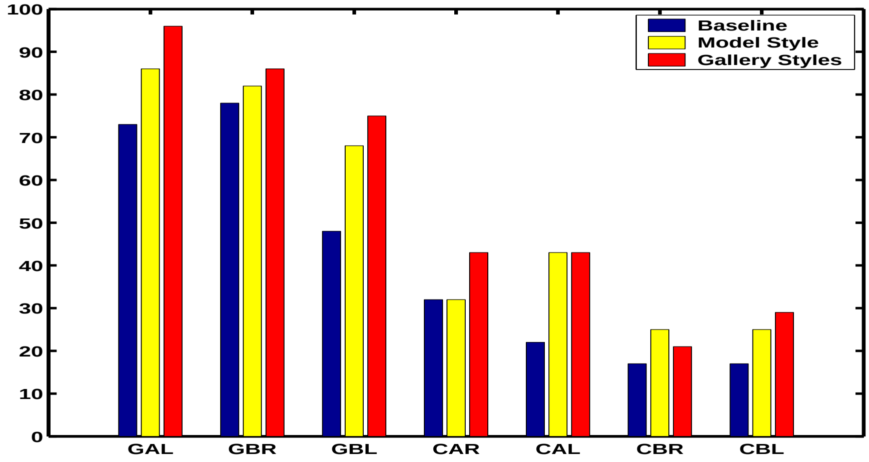


Figure 8.9: Recognition result

this experiment we use NIST-USF Gait database [101] to evaluate performance of gait recognition with continuous variation of the view due to the elliptical course that people used in capturing the database. We arbitrary select 28 people for a preliminary evaluation. We choose GAR, which is the gait sequence in grass surface, shoes type A, and right camera sequences, as a gallery set and tested by seven probe sets with variants in view, shoe and surface. Seven cycles were detected from the gallery sets and the probe sets. Three representative cycles of different views were selected from each sequence of gallery sets to learn the model.

For recognition we evaluated two classifiers for each estimated gait-style parameter for each test cycle: nearest style class mean (Model Style) and nearest neighbor style (Gallery Styles). In both cases, we used majority vote from different test cycles to determine final person identification. Results are shown in Table 8.3 and Fig. 8.9. Table 8.3 also shows recognition results reported in baseline evaluation [112] and recognition results reported using HMM by Kale *et al* in [65].

8.2 Carrying Object Detection

Iterative procedure to estimate shape style and body pose using pose preserving generative model allows estimation of outlier in addition to accurate body pose and shape style estimation. The model is also used for hole filling in the background-subtracted silhouettes using mask generated from the dynamic shape model. By iterative analysis of outlier and hole filling in the sequence of visual input, we can detect carry object efficiently.

8.2.1 Pose Preserving Dynamic Shape Models

When we know the state of the decomposable generative model, we can synthesize the corresponding dynamic shapes. For given body pose parameter, we can reconstruct best fitting shape by estimating style and view parameter with preserving the body pose. Similarly, when we know body pose parameter and view parameter, we can reconstruct best fitting shape by estimating style parameter with preserving view and body pose. If we want to synthesize new shape at time t for a given shape normalized input $\boldsymbol{y}_t$, we need to estimate the body pose $\boldsymbol{x}_t$, the view $\boldsymbol{v}$, and the shape style $\boldsymbol{s}$ which minimize the reconstruction error

$$E(\boldsymbol{x}_t, \boldsymbol{v}, \boldsymbol{s}) = || \, \boldsymbol{y}_t - \boldsymbol{\mathcal{A}} \times \boldsymbol{v} \times \boldsymbol{s} \times \psi(\boldsymbol{x}_t) \, || \; . \tag{8.15}$$

We assume that the estimated optimal style can be written as a linear combination of style vectors in the training data. Therefore, we need to solve for linear regression weights α such that $\boldsymbol{s}^{est} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$ where each $\boldsymbol{s}^k$ is one of the K_s shape style vectors in the training data. Similarly for the view, we need to solve for weights β such that $\boldsymbol{v}^{est} = \sum_{k=1}^{K_v} \beta_k \boldsymbol{v}^k$ where each $\boldsymbol{v}^k$ is one of the K_v view class vectors. This Eq. 8.15 can be solved using deterministic annealing procedure presented in Sec. 4.5.

8.2.2 Iterative Carrying Object Detection

We can detect carrying object by iterative estimation of outlier using the generative model that can synthesize pose-preserving shape. In order to achieve better alignment in normalized shape representation, we performed hole filling and outlier removal for the extracted shape.

Hole Filling

We fill holes in the background-subtracted shape to attain more accurate shape representation. When the foreground color and the background color are the same, most of the background subtracted shape silhouette has holes inside the extracted shape. This causes inaccurate description of shape in signed distance function. Hence holes inside shape result in errors in the estimation of best fitting model. It can also induce misalignment of shape as the hole can shift center of gravity for the horizontal axis alignment.

From the signed distance representation, we can generate a mask to represent inside of the shape corresponding to estimated style, view, and body pose. We can use the mask to fill holes for the original shape. The mask can be generated by thresholding generated signed distance shape representation as

$$h(x)_{hole\,mask} = \begin{cases} 1 & d_c(x) \geq d_c^{TH_{hole}} \\ 0 & \text{otherwise} \end{cases}, \tag{8.16}$$

where $d_c^{TH_{hole}} \geq 0$ is threshold value for inner shape to create mask to fill hole. If the threshold value is zero, the mask will be the same as the silhouette image generated by dynamic shape model given style, view and configuration. As we don't know the exact shape style, view and configuration at the beginning, and the hole causes misalignment, we start from large threshold value, which generate a small mask of inner shape and robust to misalignment. We reduce threshold value as parameter estimation gets more accurate. The hole filling operation can be described by $\boldsymbol{y}_{hole\ filling} = z\left(\mathtt{bin}(\boldsymbol{y}) \oplus \boldsymbol{h}(\boldsymbol{y}^{est})\right)$, where $\oplus$ is logical `OR` operator to combine extracted foreground silhouette and mask area, $\mathtt{bin}(\cdot)$ converts signed distance shape representation into binary representation, and $z(\cdot)$ convert binary representation into signed distance representation with threshold. Fig. 8.11 shows initial shape normalized silhouette with holes (a), best estimated shape model (b) which is generated from the generative model with style and view estimation and configuration search, and hole mask (c) when $d_c^{TH_{hole}} = 3$, and new hole filled shape (d). We can improve the best matching shape even initial inaccurate shape extraction for given shape style and view by excluding mask area in the computation of similarity measurement for generative samples. By re-alignment of shape and re-computing of shape representation after hole filling provide better shape description for next step.

Carrying Object Detection

Carrying objects are detected by estimating outlier from best matching normal dynamic shape and given input shape. Outliers of shape silhouettes in carrying objects are the mismatching part in input shape compared with best matching normal walking shape. Carrying objects are the major source of mismatching when we compare with normal walking shape even though other factors such as inaccurate shape extraction in background subtraction, shape misalignment cause mismatches. For accurate detection of carrying object from outlier, we need to

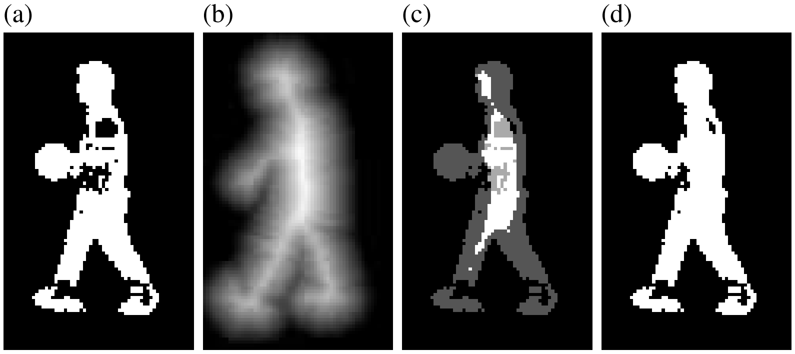


Figure 8.10: Hole filling using mask from best fitting model : (a) Initial normalized shape with hole. (b) Best matching shape from generative model. (c) Overlapping with initial silhouette and mask from best matching shape by threshold. (d) New shape with reduced hole.

remove other source of outlier such as hole and misalignment in shape. Hole filling and outlier removal are performed iteratively to improve hole for better alignment and estimation of shape.

We gradually reduce threshold value for outlier detection to get more precise estimation of outlier progressively. The mismatching error $e(x)$ is measured by Euclidian distance between signed distance input shape and best matching shape from dynamic shape model

$$e_c(x) = ||z_c(x) - z_c^{est}(x)|| \ . \tag{8.17}$$

The error $e(x)$ increase linearly as the outlier goes away from the matching shape contour due to signed distance representation. By thresholding the error distance, we can detect outlier.

$$O(x)_{outlier\,mask} = \begin{cases} 1 & e_c(x) \geq e_c^{TH_{outlier}} \\ 0 & \text{otherwise} \end{cases} , \tag{8.18}$$

At the beginning, we start from large $e_c^{TH_{outlier}}$ value and we reduce the value gradually. Whenever we detect outlier, we remove the outlier and perform realignment to remove alignment artifact due to outlier. For example, given signed distance input shape (e), we measure mismatching error (f) by comparing with best matching shape (b). Outlier is detected (g) with given threshold value $e_c^{TH_{outlier}} = 5$, and new shape for next iteration is generated by removing outlier (h) in Fig. 8.11. This outlier detection and removal are combined with hole filling as both of them help accurate alignment of shape and estimation of best matching style, view and body pose.

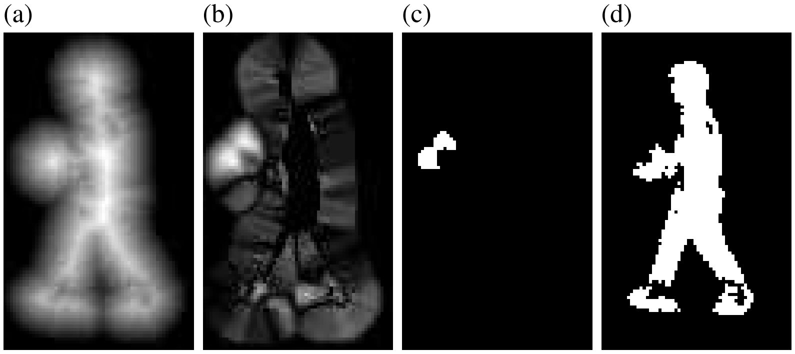


Figure 8.11: Outlier detection and removal: (a) Initial normalized shape for outlier detection with signed distance representation. (b) Euclidian distance between best matching model from the generative model and input shape with signed distance representation. (c) Detected outlier with threshold value $e(x) \geq 5$. (d) New shape after removing outlier.

Iterative Estimation of Outlier with Hole Filling

An iterative estimation of outlier, hole filling, outlier removal, and estimation of shape style, view and configuration is performed with threshold value control. The threshold value for hole filling and the threshold value for outlier detection need to be decreased to get more precise in the outlier detection and hole filling in each iteration. In addition, we control the number of sample to search body pose for estimated view and shape style from small number to large number. At the initial stage, as we cannot reach accurate estimation of body pose due to inaccuracy of shape style and view, we can use small number of sample along the manifold with equally distant samples. As the estimation progress, we estimate more accurate estimation of body pose with increased number of samples to compare with given input shape. We summarize the iterative estimation as follows:

```
Input: image $\boldsymbol{y}$, view class means $\boldsymbol{v}^k$, style class means $\boldsymbol{s}^k$, core tensor $\boldsymbol{\mathcal{A}}$

Initialization:  • initialize $T_v$ and $T_s$
    • initialize $\alpha$ and $\beta$ to uniform weights
    • Compute initial $\boldsymbol{s} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$, $\boldsymbol{v} = \sum_{k=1}^{K_v} \beta_k \boldsymbol{v}^k$
    • initialize sample num $N_{sp}$
    • initialize $d_c^{TH_{hole}}$, $e_c^{TH_{outlier}}$

Iterate:  • Compute coefficient $\boldsymbol{B} = \boldsymbol{\mathcal{A}} \times \boldsymbol{s} \times \boldsymbol{v}$
```

- Estimate body configuration: 1-D search for x that minimizes $E(x) = ||y - B\psi(x)||$
- estimate new view factor
  - Compute $p(y|x, s, v^k)$
  - Update view weights $\beta_k = p(v^k|y, x, s)$
  - Estimate new v as $v = \sum_{k=1}^{K_v} \beta_k v^k$
- Update coefficient $B = \mathcal{C} \times s \times v$
- Estimate body configuration: 1-D search for x that minimizes $E(x) = ||y - B\psi(x)||$
- estimate new style factor
  - Compute $p(y|x, s^k, v)$
  - Update style weights $\alpha_k = p(s^k|y, x, v)$
  - Estimate new s as $s = \sum_{k=1}^{K_s} \alpha_k s^k$
- Generate N_{sp} samples $\boldsymbol{y}_i^{sp}\ \boldsymbol{b}_i, i = 1, \cdots, N_{sp}$
  - Coefficient $\boldsymbol{C} = \boldsymbol{\mathcal{A}} \times \boldsymbol{s} \times \boldsymbol{v}$
  - embedding $\boldsymbol{b}_i = g(\beta_i),\ \beta_i = \frac{i}{M_{sp}}$
- Generate hole filling mask $\boldsymbol{h}_i = h(\boldsymbol{y}_i^{sp})$
- Update input with hole filling $\boldsymbol{y}_{hole\ filling} = z\left(\mathtt{bin}(\boldsymbol{y}) \oplus \boldsymbol{h}_i(\boldsymbol{y}^{est})\right)$
- Estimate best fitting shape with hole filling mask: 1-D search for $\boldsymbol{y}^{est}$ that minimizes $E(\boldsymbol{b}_i) = ||\boldsymbol{y}_{hole\ filling} - \boldsymbol{h}_i\left(\boldsymbol{C}\psi(\boldsymbol{b}_i)\right)||$
- Compute outlier error $e_c(x) = ||\boldsymbol{y}_{hole\ filling} - \boldsymbol{y}^{est}(x)||$
- Estimate outlier $\boldsymbol{o}_{outlier}(x) = e_c(x) \geq e_c^{TH_{outlier}}$

Update:

- reduce $d_c^{TH_{hole}}, e_c^{TH_{outlier}}$
- increase N_{sp}
- reduce T_v, T_s

At the end of the iteration we reach best estimation of body pose with view and shape style estimation. Based on the best matching shape, we compute the outlier from the initial source after re-centering initial source based on history of pose alignment.

8.2.3 Experimental Results

We evaluated our method using two gait-database. One is from CMU Mobo dataset and the other is our own dataset for multiple view gait sequence. Robust outlier detection in spite

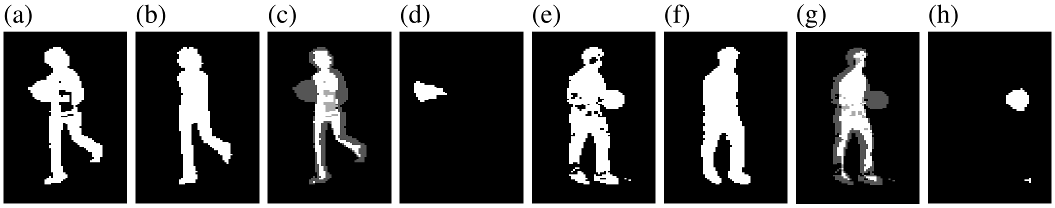


Figure 8.12: Outlier detection in different view: (a) Initial normalized shape for outlier detection. (b) The best fitting model from the generative model. (c) Overlapping initial input and hole filling mask at the last iteration. (d) Detected outlier. (e) (f) (g) (h) : Another view in different person

of hole in the silhouette images was shown clearly in CMU database. We collected our own dataset to show carrying object detection in continuous view variations.

Carrying Ball Detection from Multiple Views

The CMU Mobo database contains 25 subjects with 6 different views walking on the treadmill to study human locomotion as a biometric [48]. The database provides silhouette sequence extracted based on one background image. Most of the sequences have holes in the background subtracted silhouette sequences. We collected $12(= 4 \times 3)$ cycles to learn dynamic shape models with view and style variations from normal slow walking sequences of 4 subjects with 3 different views. For the training sequences, we corrected holes manually. Fig. 8.12 shows detected carrying objects in two different views from different people. The initial normalized shape has holes with a carrying ball (a)(e). Still the best fitting shape models recover correct body pose after iterative estimations of view and shape style with hole filling and outlier removal (b)(f). Fig. 8.12 (c)(g) show examples of generated masks during iteration for hole filling. Fig. 8.12 shows example outlier detected at the end. In Fig. 8.12 (h), the outlier in bottom right corner comes from the inaccurate background subtraction outside the subject, which cannot be managed by hole filling. The verification routine based on temporal characteristics of the outlier similar to [8] can be used to exclude such a outlier from detected carrying objects.

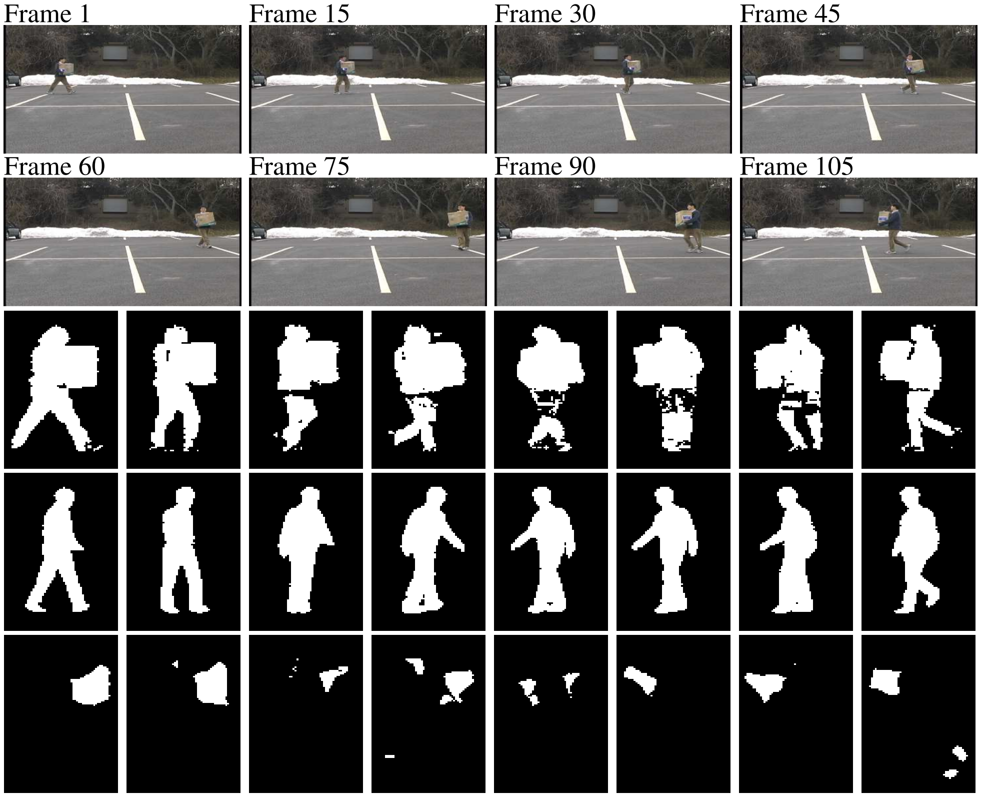


Figure 8.13: Outlier detection in continuous view variations: First row: Input image. Second row: Extracted silhouette shape. Third row: Best matching shape. Fourth row: Detected carrying object

Carrying Object Detection with Continuous View Variations

We collected 4 people with 7 different views to learn the pose preserving shape model of normal walking for detection of carrying object in continuous view variations. In order to achieve reasonable multiple views interpolation, we captured normal gait sequence on the treadmill with the same height camera position in the lab. The test sequence is captured separately in outdoor using commercial camcorder. Fig. 8.13 and Fig. 8.13 shows an example sequence of carrying object detection in continuous change of walking direction. The first row shows original input images from the camcorder. The second row shows normalized shape after background subtraction. We used the nonparametric kernel density estimation method for per-pixel background models, which is proposed in [35]. The third row shows best matching shape estimated after hole filling and outlier removal using dynamic shape models with multiple views. The fourth row shows detected outlier. Most of the dominant outlier comes from the carrying object.

8.3 High Resolution Facial Expression Control from Video Sequence

Human faces can express not only basic emotions such as anger, surprise, or happiness, but also subtle thoughts or emotions. If we take the variations of a smile as an example, they communicate not only happiness but many other cognitive/emotional states: a smile of enjoyment, a scornful smile, a pleased smile, a flirtatious smile, a heart-warming smile, a smirk, and so on. In order to analyze the subtle differences between individuals, decomposable nonlinear generative models are proposed to model subtle motions and dynamic texture changes of different people. For this purpose, an empirical kernel map along with an embedded manifold and a projection, is introduced to represent the nonlinear mapping for each cycle of facial motions. The subtle differences of facial motions across different persons and expressions are represented in the projection. Through a multi-linear analysis of these linear projections, we can decompose the nonlinear mappings into two main factors: the *personal style*, individual characteristics of expressions, and the *expression type*, subtle variations in expressions. Consequently, new facial expressions can be generated by using different personal style and expression type factors, along with the common embedded manifold, which encodes the temporal information of facial expressions. We also present a performance-driven approach to create high resolution facial expressions based on an exemplar video sequence.

8.3.1 System Overview

Our high resolution facial expression synthesis system includes five main components: data acquisition, facial motion tracking, modeling facial expressions with decomposable generative models, estimation of facial expression control parameters, and synthesis of high resolution facial expressions. In the data acquisition stage, we collect both high resolution dynamic range data of facial expressions and 2D video sequences separately for each subject. Both 2D and 3D facial expression sequences are captured at high frame rates from multiple people with several expression types. In order to establish correspondences between different frames within one sequence and between different sequences, we employed a high resolution 3D tracking method using harmonic maps [144] to extract detailed facial motions with subtleties from dynamic range data, while low resolution facial motions are extracted from video sequences using a 2D

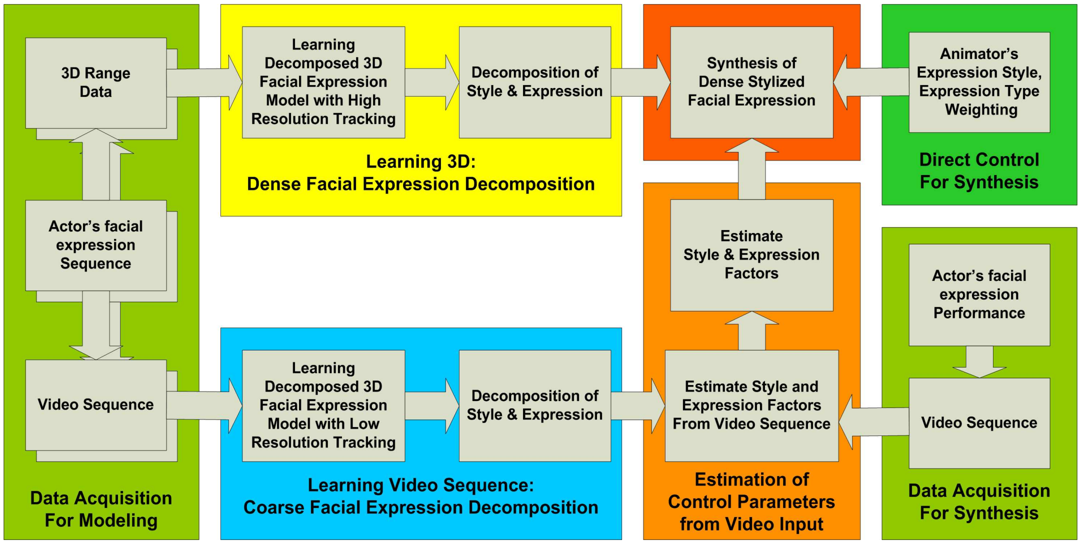

3D Range Data
Actor's facial expression Sequence
Video Sequence
Data Acquisition For Modeling
Learning Decomposed 3D Facial Expression Model with High Resolution Tracking
Decomposition of Style & Expression
Learning 3D: Dense Facial Expression Decomposition
Learning Decomposed 3D Facial Expression Model with Low Resolution Tracking
Decomposition of Style & Expression
Learning Video Sequence: Coarse Facial Expression Decomposition
Synthesis of Dense Stylized Facial Expression
Estimate Style & Expression Factors
Estimate Style and Expression Factors From Video Sequence
Estimation of Control Parameters from Video Input
Animator's Expression Style, Expression Type Weighting
Direct Control For Synthesis
Actor's facial expression Performance
Video Sequence
Data Acquisition For Synthesis

Figure 8.14: Components of high resolution facial expression synthesis system from video sequences

tracking method based on 2D contours and 3D deformable models.

Because the tracking results extracted from 2D video sequences and from dynamic range data have different levels of detail, we derive from these two sources of facial motion data, two generative models with different resolutions and the same kinds of state decomposition. In order to analyze the subtle differences between individuals, we propose decomposable nonlinear generative models that model subtle motions of different people. For this purpose, an empirical kernel map, along with the embedding manifold and the linear projection, is introduced to represent the nonlinear mapping for each cycle of facial motions. However, since both 2D and 3D tracking data are extracted from similar facial expressions, the same conceptual motion manifold can be applied to both data. Consequently, the differences in high resolution facial motions with subtleties and low resolution facial motions are encoded in the linear projections following the same empirical kernel map. In order to decompose the nonlinear mapping into multiple expression factors, a multi-linear analysis is performed on high resolution and low resolution tracking data separately. This analysis provides a decomposition of each personal style and expression type using multiple expression bases. Although these expression bases associated to high resolution facial motions with subtleties and low resolution facial motions are different from each other because of different levels of detail, *an important observation is that a new expression vector estimated from different resolution tracking data shares a similar* *distance to each expression basis in all levels of detail.* It also implies that the normalized expression bases provided by our nonlinear decomposition algorithm are invariant to the levels of detail and different vector dimensions after the kernel mapping.

Finally, new high resolution stylized facial expressions can be generated using the nonlinear generative models with two main factors: "personal style" and "expression type". In particular, new facial expressions can be synthesized by two approaches: (1) by changing directly the weighting of personal style and expression type or (2) by estimating the weighting of personal style and expression type from an exemplar video sequence of a target subject's facial expression.

8.3.2 Modeling and Analyzing Facial Expressions

A major issue in creating facial expression animation is how to model and control the dynamics in facial motions, which often include nonlinear deformations. In addition, the facial motions usually do not take place in uniform speed and depend on both personal styles and expression types. We represent facial motions by motion fields of a high dimensional generic model to capture detailed expression motion, and each frame is formed in a high dimensional vector by collecting the vertex displacements. To begin with, a low dimensional representation of facial motions is derived using conceptual motion manifold embedding. Then, kernel mappings are utilized to capture nonlinear characteristics in the facial motions. Multilinear analysis of the nonlinear mapping coefficients, which encode different personal styles and expression types, provides decomposable nonlinear generative models for the facial motions with compact state parameterizations.

Based on sample data collected for tracking facial expressions with expression change from a neutral to a specific target expression and to neutral again from each person and expression type, we can learn a nonlinear generative mapping between the embedding space and the original facial motion. Given a facial motion sequence $\boldsymbol{Y}^{se} = [\boldsymbol{y}_1^{se}\ \boldsymbol{y}_2^{se} \cdots \boldsymbol{y}_{N_{se}}^{se}]^T$, where N_{se} is the number of captured motion frames for the sequence with style s and expression e, we can embed such sequence temporally at equidistance points on a unit circle such that $\boldsymbol{x}_i^{se} = [\cos(2\pi i/N_{se})\ \sin(2\pi i/N_{se})], i = 1 \cdots N_{se}$. With each entire sequence and its embedding $\boldsymbol{X}^{se} = [\boldsymbol{x}_1^{se}\ \boldsymbol{x}_2^{se} \cdots \boldsymbol{x}_{N_{se}}^{se}]^T$, we can learn a nonlinear mapping function $f^{se}(\boldsymbol{x})$ that satisfies $f^{se}(\boldsymbol{x}_i) = \boldsymbol{y}_i^{se}$, $i = 1 \cdots N_{se}$. Using empirical kernel map as described in Sec. 3.3, we learn nonlinear mapping of the form

$$f^{se}(\boldsymbol{x}) = \boldsymbol{B}^{se} \cdot \psi(\boldsymbol{x}), \tag{8.19}$$

where $\boldsymbol{B}$ is a $d \times N$ coefficient matrix and $\psi(\cdot) : \mathbb{R}^l \to \mathbb{R}^N$ is a kernel mapping.

For a given kernel $\psi(\boldsymbol{x})$, the matrix $\boldsymbol{B}^{se}$ captures the facial motion characteristics for expression style s and type e. Given facial motion sequences with N_s personal styles and N_e expression types, we obtain $N_s \times N_e$ mappings. By converting each mapping matrix $\boldsymbol{B}$ into the corresponding mapping coefficient vector $\boldsymbol{b}^{se}$, by column stacking, the collection of mapping coefficients can arranged into a high order tensor of personal styles and expression types.

Modeling Facial Expressions

Facial motion can be described by vertex movements in each frame from high resolution 3D tracking and from low resolution video sequence tracking as we preserve one-to-one intra-frame correspondences in model-based tracking. Let $\boldsymbol{v}_t \in R^{3N \times 1}$ be locations of 3D points at time instance t representing N facial nodal points in a 3-dimensional space, where N is the number of nodal points in a dense generic facial model. The trajectory of the 3D nodal points is the combination of global transformation and facial motion and varies for each person and expression. The captured vertex data is a combination of global transformation, person face geometry, and facial motion. It can be described for the personal style s and the expression type e as

$$\boldsymbol{v}_t^{se} = T_{\boldsymbol{\alpha}_t} \boldsymbol{y}_t^{se} = T_{\boldsymbol{\alpha}_t}(\boldsymbol{y}_{t_0}^{se} + \boldsymbol{m}_t^{se}), \tag{8.20}$$

where $T_{\boldsymbol{\alpha}_t}$ is the global transformation due to head motion at time t, $\boldsymbol{y}_t^{se}$ is the face nodal point location at time t after normalization for global transformation with dimension d, including motion and geometry, $\boldsymbol{y}_{t_0}^{se}$ is the facial geometry at the initial frame $\boldsymbol{m}_t$ is the displacement of vertex points from initial geometry. Since for each person, the expression sequence was collected to start from neutral expressionwe assume that the initial frame $\boldsymbol{y}_{t_0}^{se}$ is the person geometry. $\boldsymbol{y}_t$ is the combination of person geometry and facial motion at time t. If we need to model only facial motion, we can use $\boldsymbol{m}_t$. When we are interested in both facial motion and person geometry, we use vertex movement $\boldsymbol{y}_t$ after normalizing global transformation $T_{\boldsymbol{\alpha}_t}$.

8.3.3 Estimations of Personal Style, Expression Type, and Motion Configuration

When we know the state of the decomposable facial expression model, we can synthesize the corresponding facial expressions. For a given tracking data $\boldsymbol{y}_t$, we need to estimate the motion configuration $\boldsymbol{x}_t$, the expression type $\boldsymbol{e}$, and the personal style $\boldsymbol{s}$ which minimize the reconstruction error

$$E(\boldsymbol{x}_t, \boldsymbol{e}, \boldsymbol{s}) = || \boldsymbol{y}_t - \boldsymbol{\mathcal{C}} \times \boldsymbol{e} \times \boldsymbol{s} \times \psi(\boldsymbol{x}_t). || \tag{8.21}$$

in order to know the control parameter for the given data. We assume that the estimated optimal personal style can be written as a linear combination of style vectors in the training data. The personal style vectors are orthogonal to each other (as are the expression type vectors) thus any expression style can be uniquely represented by their combination. Therefore, we need to solve for linear regression weights α such that $\boldsymbol{s}^{est} = \sum_{k=1}^{K_s} \alpha_k \boldsymbol{s}^k$ where each $\boldsymbol{s}^k$ is one of the K_s personal style vectors in the training data. Similarly for the expression type, we need to solve for weights β such that $\boldsymbol{e}^{est} = \sum_{k=1}^{K_e} \beta_k \boldsymbol{e}^k$ where each $\boldsymbol{e}^k$ is one of the K_e expression type vectors. An iterative procedure described in Sec. 4.5, is used to estimate $\boldsymbol{x}_t, \boldsymbol{e}, \boldsymbol{s}$ from given input $\boldsymbol{y}_t$ similar to [78]. We applied the estimation procedure to low resolution video sequence tracking to estimate the state of high resolution 3D model.

8.3.4 Control Parameter Estimation from Video Sequence

We can achieve approximate estimation of the personal style and the expression type parameters for low resolution facial motion model from a 2D video sequence. There estimates can then be applied to the control of the corresponding high resolution 3D facial motions. The low resolution tracking results from video sequences show distinguishable variations in different people and in different expressions even though it does not capture subtle expression details in the expression such as wrinkles. The nonlinear decomposable model for low resolution tracking from 2D texture images, which are corresponding to every range data used for high resolution tracking, establishes new basis for the low resolution 3D tracking from video for the same person and expression with the same number of sequence used for decomposition of dense 3D tracking. Even though the basis for the personal style and the expression type is different between high resolution facial motion model and low resolution facial motion model, the relative distances in the basis for the estimated personal style and the expression type are preserved. Therefore, we can use the weighting factor, which represents the similarity of the estimated expression by the known personal style and expression type, which is proportional to the exponential of distance from estimated style vector and expression vector to the known style and expression vector, to control high resolution facial expression generative models with estimated weight parameters.

8.3.5 Experimental Results

We demonstrate the performance of our framework through analysis and synthesis of several types of smiles that often happen on human faces. In order to acquire a small database of facial expressions, we invited two actors and one actress to perform three different types of smile: a) soft affectionate (SA) smile, b) coy flirtatious (CF) smile, and c) devious smirk(DS). In addition the same instructions are provided to achieve relatively consistent facial expressions across actors/actresses. Afterwards, we learned a nonlinear generative model for three subjects ($N_s = 3$) with three different types of smile expressions ($N_e = 3$) based on high resolution tracking results from range data.

To synthesize high resolution stylized facial motions in different expressions, we only need to estimate the weighting of personal styles and expression types, which allow us to generate new facial motions from the decomposable nonlinear generative model. Given certain personal style and expression type vectors, we can generate new facial expressions from generative models by tensor multiplications. We provide two methods to synthesize new stylized facial expressions: (1) direct control of the weighting parameters and (2) a video-driven approach.

Direct control of the weighting parameters

We can generate new personal styles and expression types by a weighted linear combination of the existing personal style and expression type vectors that are provided by the decomposable generative model learned from the collected sequences. i.e.

$$\boldsymbol{e}^{new} = w_{e_1}\boldsymbol{e}_1 + w_{e_2}\boldsymbol{e}_2 + \cdots + w_{e_{N_e}}\boldsymbol{e}_{N_e}, \tag{8.22}$$

where $\sum_{1}^{N_e} w_i = 1$. As each weight of expression type is proportional to the conceptual similarity of new expression to each given expression type used for modeling facial motions, a user can turn the expression weight efficiently to generate new type of expression. The user can tune the person style parameter similarly based on similarity of the new sequence to the people used for modeling. The new personal style and expression type vector will define a new linear projection $\boldsymbol{B}^{new}$ and generate nonlinear facial motions after kernel mapping. The linear combination can be adjusted intuitively even by non-expert user based on similarity weighting of the target sequence and the basis personal styles and expression types. In addition, as the proposed model represent facial motions in decomposable generative framework, the style weight, the expression weight, and dynamic factor (configuration embedding) can be controlled independently.

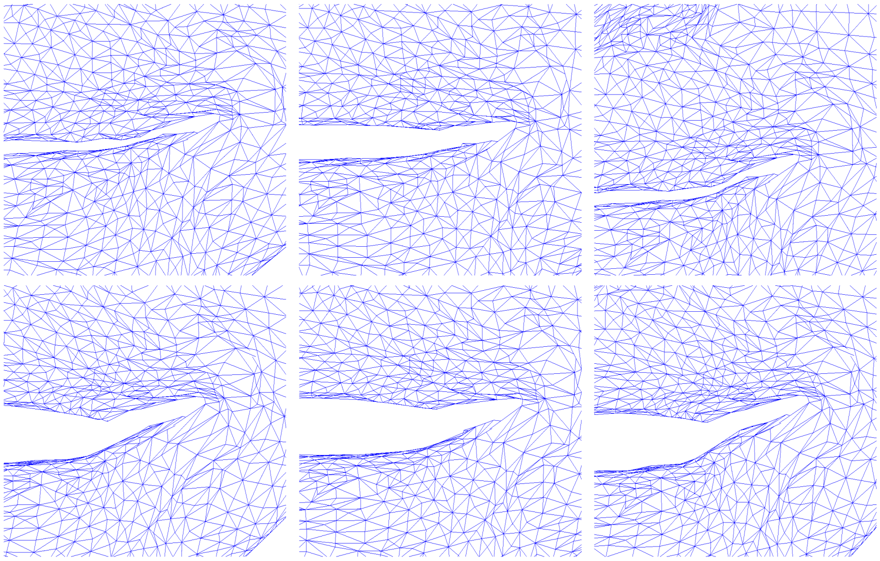


Figure 8.15: Comparison of configuration interpolation on embedding manifold and linear interpolation: Rows: $\frac{1}{5}T$ and $\frac{2}{5}T$, where T is the total frame number. Left column: original facial motions. Middle column: linear interpolation of intermediate expression based on the beginning and peak expression frame. Right column: synthetic results from the embedding manifold.

Fig. 8.15 compares the synthetic results of our method to the results by the linear interpolation. As we can see, the linear combination of the beginning and the peak expression frame cannot generate subtle motions in the intermediate state captured using high resolution tracking (in the middle column). However, our proposed nonlinear generative model with manifold embedding and kernel map counting dynamics can synthesize subtle details in the intermediate expressions (right column) closer to the original motions (left column). Fig. 8.16 shows a example of morphing on both the personal style and the expression type. It starts with one subject's soft affectionate smile with certain expression type and personal style and then changes to another subject's devious smirk with a smooth transition in the middle frames. Fig. 8.17 shows an example of transferring personal styles. Even in the same face geometry, you can still see the different subtleties caused by different personal style.

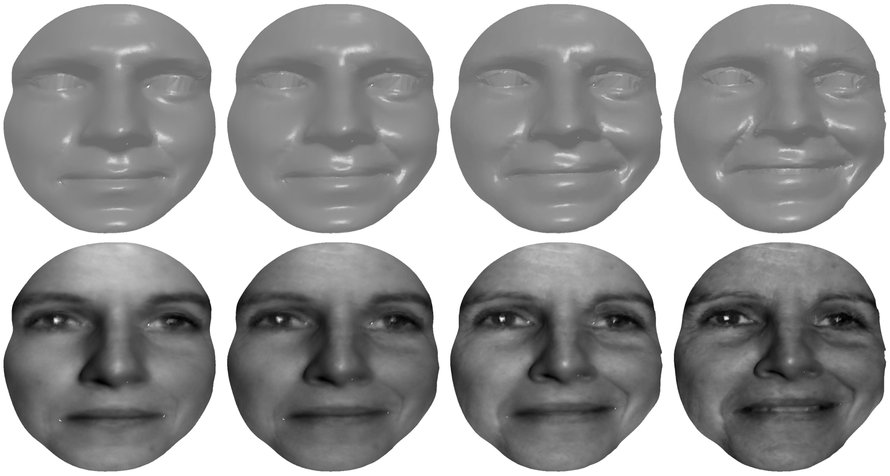


Figure 8.16: An example of morphing on both the personal style and the expression type: A morphing from the soft affectionate (SA) smile of one subject to the devious smirk (DS) of another subject.

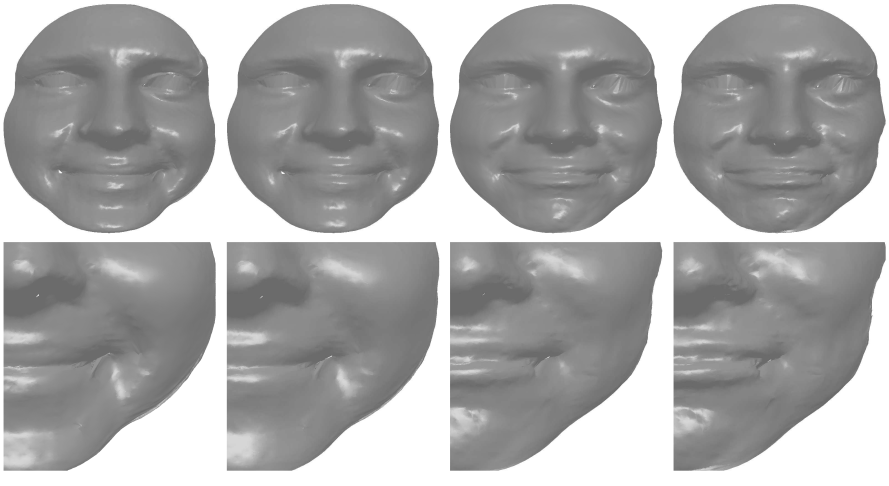


Figure 8.17: Synthesizing new facial personal styles: First column: Subject A's soft affectionate smile, Forth column: Subject B's soft affectionate smile, Second column: 75% style A + 25% style B, and Third column: 25% style A + 75% style B.

Video-driven approach

Another scheme to control high resolution facial expressions is a video-driven approach, by estimating the weighting of personal styles and expression types from an exemplar video sequence of a target subject's facial expression. The 2D contour tracking can be done 60 frames/sec and low resolution 3D facial motion tracking from the 2D contour and captured image by 5 frames/sec in C++ implementation. Using the low resolution 3D tracking data, we estimate the personal styles, expression type, and motion configuration parameters for every frame. Then, we synthesize new facial expressions based on the estimated parameters using the high resolution facial expression models.

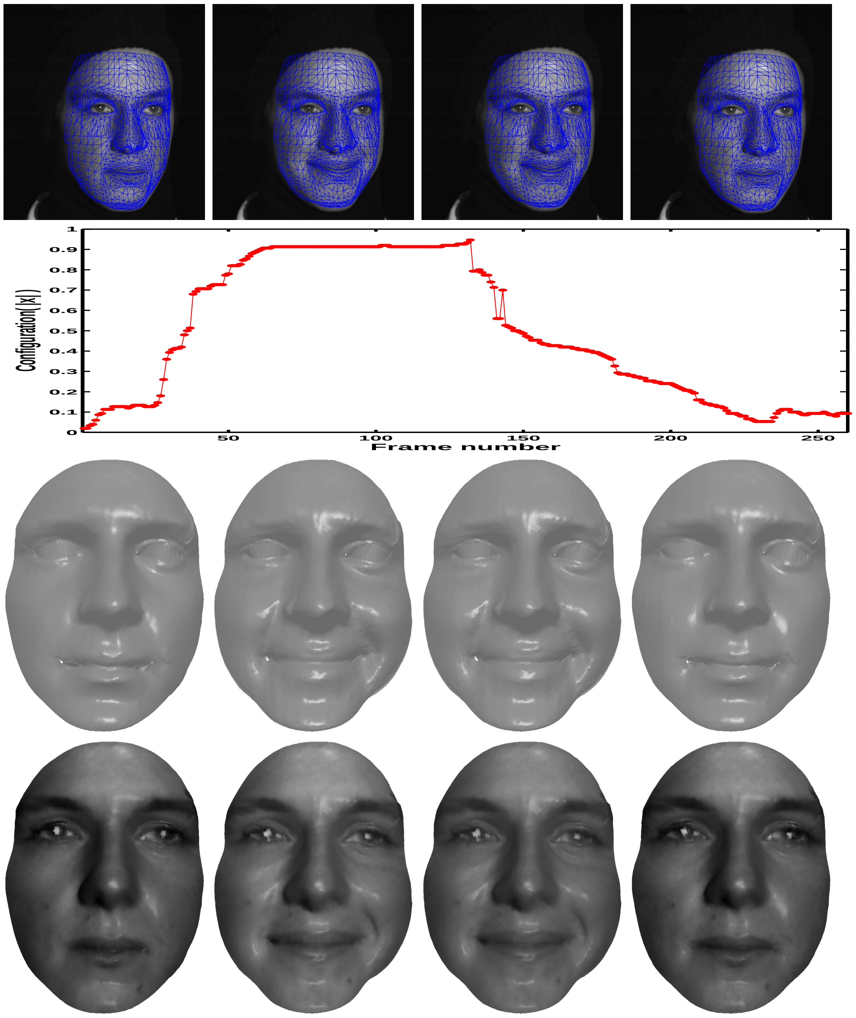


Figure 8.18: A synthetic facial expression of a subject in the training database, based on a video sequence of an expression that was not used in the training stage.

Fig. 8.18 shows a synthetic dense 3D facial expression sequence based on an input 2D video sequence. Even though the new expression has different temporal characteristics compared to the sequences used for facial motion modeling, the estimated configuration parameters can capture the temporal differences by the new configuration sequences. Fig. 8.19 shows the estimation of the personal style and the expression type of a new subject. The new subject's style is represented by a weighted combination of training subjects' styles. Expression type parameters are accurately estimated for a correct devious smirk. In addition, the head motions extracted from the low resolution 3D tracking are integrated in the resulting global transformation T_α, which makes the synthesized facial expression more realistic.

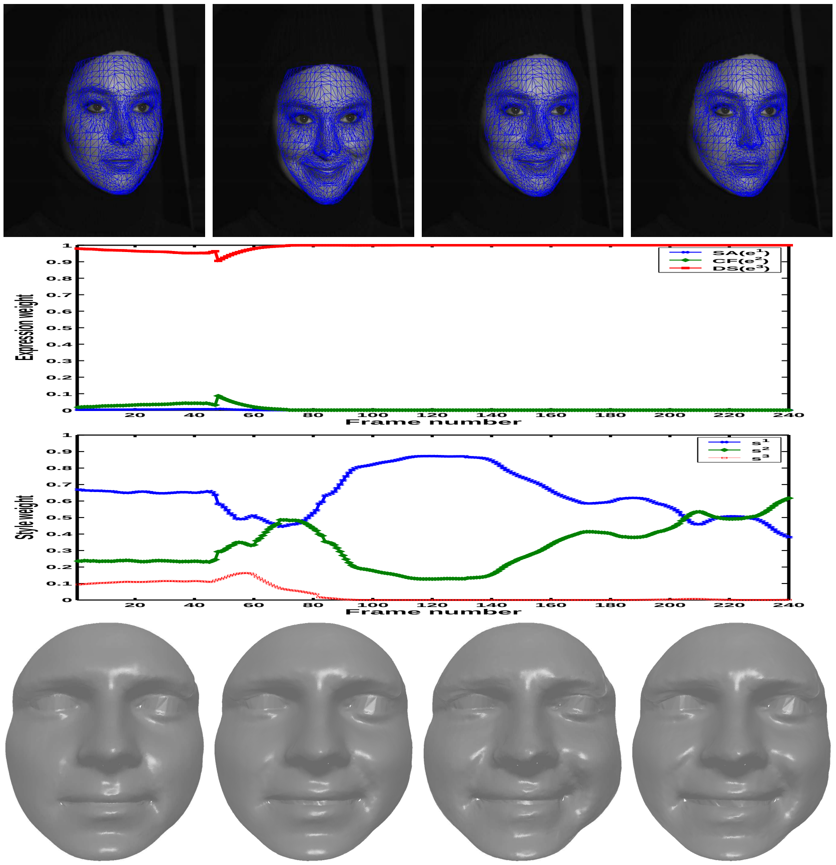


Figure 8.19: Performance-driven animation of the high resolution facial expression motion from video sequence tracking: From the input video sequence with low resolution tracking (in the first row), we can estimate the weighting of expression types (second row) and personal styles (third row). Using the estimated weighting, we can synthesize high resolution facial expression animation (in the forth row).

Fig. 8.20 shows synthesized high resolution images from two different subtle expressions. Even though the images are similar, the estimated personal style and expression type parameters distinguish subtle difference and we can generate high resolution facial expressions with subtle differences. In video sequence, we compare two generated motions. On the left column, we see the original low resolution video sequence, which is used to capture the motion. On the middle column we see the expression that was generated using only the low-resolution captured motion. On the right column we see the same sequence enhanced by subtle details generated by our high resolution decomposable generative model.

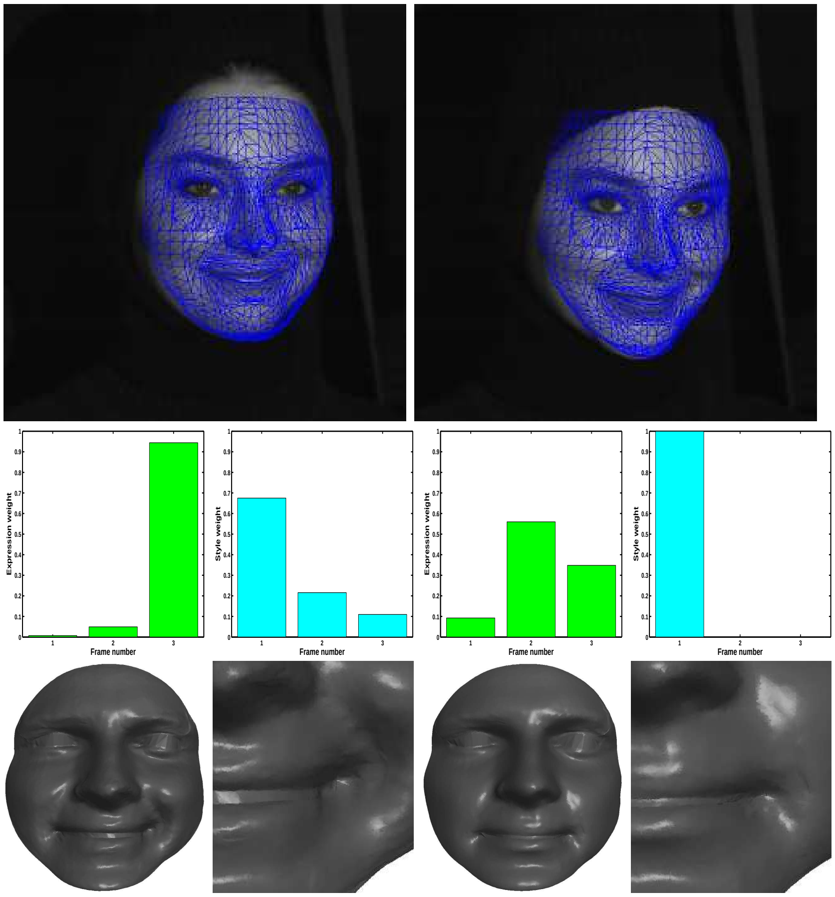


Figure 8.20: Synthesis of expressions with subtle differences: First row : an input video sequence with low resolution tracking. Second row : Estimated expression type weight (right) and style weight (left) for each video image. Third row : Synthesized high resolution expressions (left) and details around mouth corner(right) using estimated style and expression type from each video image.

Chapter 9

Conclusion

We presented a novel framework for modeling dynamic shape and appearance of articulated human motions. We introduced a framework for learning global representations of dynamic shape and dynamic appearance manifolds. The framework is based on using nonlinear manifold learning to achieve an embedding of the global deformation manifold, which preserves its the geometric structure. Given such embedding, a nonlinear mapping is learned from such embedded space into visual input space using RBF interpolation. Given this framework, any visual input is represented by a linear combination of nonlinear bases functions centered along the manifold in the embedded space. In a sense, the approach utilizes the implicit correspondences imposed by the global vector representation, which are only valid locally on the manifold, through explicit modeling of the manifold and the RBF interpolation where closer points on the manifold will have higher contributions than far away points.

We showed how to learn a factorized generative model that separates appearance variations from the intrinsics underlying dynamics manifold though introducing a framework for separation of style and content on a nonlinear manifold. The framework is based on decomposing the style parameters in the space of nonlinear functions that maps between a learned unified nonlinear embedding of multiple content manifolds and the visual input space. The framework yields an unsupervised procedure that handles dynamic, nonlinear manifolds. It also improves on past work on nonlinear dimensionality reduction by being able to handle multiple manifolds. The proposed framework was shown to be able to separate style and content on both the gait manifold and a simple facial expression manifold. We further extend the model to cover multiple factor variations, such as view and motion type, using multilinear tensor analysis and conceptual manifold embedding.

To model the continuous view manifold as well as the continuous body configuration manifold, we used a product manifold representation using both supervised and unsupervised manifold learning techniques. We explicitly model view manifold and body pose manifold with two orthogonal components on a torus manifold for one dimensional human motion such as gait and golf swings. As an alternative approach, we model the continuous view manifold invariant to body configuration based on an embedding of the kinematics' manifold. View manifold is parameterized by factorization of the nonlinear mapping coefficients from a common kinematics embedding to view variant sequences captured along a view circle.

We demonstrated the advantages of our generative model for high dimensional dynamic human motion analysis, tracking and synthesis by applying to several applications. Using shape style, or expression type parameters within our factorized generative models, we parameterize the characteristics of dynamic shape and appearance variations by static feature vectors such as shape style and expression type, which are invariant to body poses (dynamic components). As a result, we achieve robust gait recognition and facial expression recognition from sequences of motion or from a single image frame.

Utilizing low dimensional body configuration embedding, we formulate the human motion tracking problem as body configuration estimation on the low dimensional manifold embedding, as well as, a style factor estimation among different people. Since the framework is generative, it fits well in the Bayesian tracking framework and it provides separate low dimensional representations for each of the modeled factors. Moreover, a dynamic model for configuration is well defined since it is constrained to the low dimensional manifold representation. The low dimensional nonlinear manifold embedding preserves intrinsic constraints of human motion and achieves robust tracking results. Additionally, as we estimate variations of shape in the factorized shape style during tracking, we achieve simultaneous estimation of person identity as well as body configuration from gait video sequence.

We showed how the learned representation can be used to interpolate intermediate body poses as well as in recovery and reconstruction of the input. We extended the approach to learn mappings from the embedded motion manifold to 3D joint angle representation which yields an approximate closed-form solution for 3D pose recovery. The dynamic shape and appearance model is also applied for modeling shape deformation during facial expressions. The dynamic shape and appearance model of facial expressions combined the global shape and appearance deformation and local deformation using appearance templates, which are updated through the global shape appearance model. Proposed models are also applied for carrying object detection, emotion recognition and synthesis, high-resolution facial expression synthesis, inferring body pose, and tracking articulated human motion in video sequences.

Modeling complicated human motion is still a challenging problem even though some of our approach is directly applicable to complicated motion. One-dimensional manifolds can be explicitly modeled in a straight forward way. The complicated human motion may be represented by a combination of primitive motions. Finding primitive motions automatically, and segmenting a complicated motion into a combination of motion primitives, and analyzing and synthesizing the complicated motion based on motion primitives are challenging problems for modeling general human motion. The generalization of factorized decomposable generative model to such a complicated human motion can be useful for arbitrary action recognition, tracking for intelligent surveillance, sport video analysis and retrieval, emotion recognition and human computer interaction.

In case of multiple sequences from different people, the data lie on multiple manifolds. Given multiple motion sequences from the same type of motion in different people, an underlying common manifold among the different data sets might exist. Current manifold learning techniques fail in finding common manifold representation from multiple manifolds. In our work, we presented two manifold embedding techniques to overcome such problems: one is a unified manifold embedding learned from individual manifolds of individual sequences; and the other is a visual observation invariant manifold such as a conceptual representation, or a kinematic manifold. However, our current approach may not be optimal to represent a common manifold from multiple manifolds. Extension to simultaneous modeling of multiple manifolds is very challenging. A better representation might be found from multiple manifolds by analyzing the inter- and the intra-manifold distances.

Visual learning based on manifold embedding contributes to the fundamental research of human cognitive system and representation of dynamic motions in artificial intelligence, and is also applicable to other related domains such as medical image analysis and sequence comparison in bio-informatics.

Appendix A

Higher-order Tensor Analysis

Multilinear model is a generalization of linear model (one-factor models) and bilinear model (two-factor models) [128] into higher-order tensor decomposition (multi-factor models). It is called n-mode analysis, multimode component analysis [87], Tucker3 model, which was originally proposed and developed in [132, 67]. Higher-order singular value decomposition (HOSVD) is a generalization of SVD for higher-order tensor analysis by [72] and extended for lower dimensional approximation by higher-order orthogonal iteration method [73]. There are three important extensions of linear algebra for higher order tensor analysis using HOSVD.

First, *matrix unfolding* is defined to represent higher-order tensor into matrix form. Higher-order tensor can be expressed as a collection of regular two-dimensional matrix. Matrix unfolding of a given Nth-order tensor $\boldsymbol{\mathcal{D}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ can be defined as a two-dimensional matrix $\boldsymbol{D}_{(n)} \in \mathbb{C}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1})}$, which contains element $a_{i_1 i_2 \cdots i_N}$ at the position with row number i_n and column number j_n [1].

Second, matrix multiplication of higher-order tensor is defined. The n-mode multiplication of a higher-order tensor $\boldsymbol{\mathcal{D}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $\boldsymbol{U} \in \mathbb{C}^{J_n \times I_n}$, can be defined as an $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \cdots \times I_N)$-tensor [72] whose entries are given by

$$(\boldsymbol{\mathcal{D}} \times_n \boldsymbol{U})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} \overset{def}{=} \sum_{i_n} a_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n} \qquad \text{(A.1)}$$

Using this definition, we can express a matrix SVD decomposition $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{F}\boldsymbol{V}^T$ by tensor multiplication notation $\boldsymbol{H} = \boldsymbol{F} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}$.

[1] $j_n = (i_{n+1} - 1) I_{n+2} I_{n+3} \cdots I_N I_1 I_2 \cdots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \cdots I_N I_1 I_2 \cdots I_{n-1} + \cdots + (i_N - 1) I_1 I_2 \cdots I_{n-1} + (i_1 - 1) I_2 I_3 \cdots I_{n-1} + (i_2 - 1) I_3 I_4 \cdots I_{n-1} + \cdots + i_{n-1}$

Finally, *Nth-order SVD* of $(I_1 \times I_2 \times \cdots \times I_N)$-tensor $\boldsymbol{\mathcal{D}}$ is defined as the product

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{U}^{(1)} \times \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)} \tag{A.2}$$

, where $\boldsymbol{U}^{(n)} = \left(\boldsymbol{U}_1^{(n)} \boldsymbol{U}_2^{(n)} \cdots \boldsymbol{U}_{I_n}^{(n)}\right)$ is a unitary $(I_n \times I_n)$-matrix and $\boldsymbol{\mathcal{Z}}$ is a $(I_1 \times I_2 \times \cdots \times I_N)$-tensor whose subtensors satisfies all-orthogonality [2]and ordering [3]. The Nth-order SVD can be computed by

1. For $n = 1, \cdots, N$, compute matrix $\boldsymbol{U}^{(n)}$ in Eq. A.2 by computing the SVD of the unfolded matrix $\boldsymbol{D}_{(n)}$ and setting $\boldsymbol{U}^{(n)}$ to the left matrix of the SVD.

2. Solve for the core tensor by

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{D}} \times_1 \boldsymbol{U}^{(1)^T} \times_2 \boldsymbol{U}^{(2)^T} \cdots \times_N \boldsymbol{U}^{(N)^T} \tag{A.3}$$

We used this HOSVD in multiple component analysis in kernel space for dynamic human motion analysis.

[2] two subtensors $\boldsymbol{\mathcal{Z}}_{i_n=\alpha}$ and $\boldsymbol{\mathcal{Z}}_{i_n=\beta}$ are orthogonal for all possible values of n, α, and β: $\langle \boldsymbol{\mathcal{Z}}_{i_n=\alpha}, \boldsymbol{\mathcal{Z}}_{i_n=\beta} \rangle = 0$ when $\alpha \neq \beta$

[3] $||\boldsymbol{\mathcal{Z}}_{i_n=1}|| \geq ||\boldsymbol{\mathcal{Z}}_{i_n=2}|| \geq ||\boldsymbol{\mathcal{Z}}_{i_n=I_n}|| \geq 0$

References

[1] B. Abboud and F. Davoine. Bilinear factorisation for facial expression analysis and synthesis. *IEE Proc. Vis. Image Signal Proceess.*, 152(3), 2005.

[2] A. Agarwal and B. Triggs. 3d human pose from silhuettes by relevance vector regression. In *CVPR*, volume 2, pages 882–888, 2004.

[3] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. *Comput. Vis. Image Underst.*, 73(3):428–440, 1999.

[4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469–483, 1996.

[5] Adam Baumberg and David Hogg. Generating spatiotemporal models from examples. *Image and Vision Computing*, 14(8):525–532, 1996.

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

[7] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[8] Chiraz BenAbdelkader and Larry S. Davis. Detection of people carrying objects: A motion-based recognition approach. In *Proc. of FGR*, pages 378–383, 2002.

[9] C. BenAdbelkader, Ross Cutler, and Larry Davis. Motion-based recognition of people in eigengait space. In *Proc. of FGR*, pages 254–259, 2002.

[10] 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 *NIPS 16*, 2004.

[11] 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.*, 16(10):2197–2219, 2004.

[12] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272(5250), 1996.

[13] B. Bhanu and Ju Han. Individual recognition by kinematic-based gait analysis. In *Proc. of ICPR*, volume 3, pages 343– 346, 2002.

[14] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using view-based representation. *Int.J. Compter Vision*, pages 63–84, 1998.

[15] Randolph Blake and Maggie Shiffrar. Perception of human motion. *Annual Review of Psychology*, 58(1):47–73, 2007.

[16] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI*, 11(6):567–585, 1989.

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

[18] M. Brand. Shadow puppetry. In *Proc. of ICCV*, volume 2, pages 1237–1244, 1999.

[19] 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.

[20] C. Bregler and S. M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *ICCV*, pages 494–499, 1995.

[21] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *Proc. of ICCV*, page 624, Washington, DC, USA, 1995. IEEE Computer Society.

[22] Ya Chang, Changbo Hu, and Matthew Turk. Probabilistic expression analysis on manifolds. In *Proc. of CVPR*, pages 520–527, 2004.

[23] Chris Mario Christoudias and Trevor Darrell. On modelling nonlinear shape-and-texture appearance manifolds. In *CVPR (2)*, pages 1067–1074, 2005.

[24] Erika S. Chuang, Hrishikesh Deshpande, and Chris Bregler. Facial expression space learning. In *Proc. of PG*, pages 68–76, 2002.

[25] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. of CVPR*, pages 44–51, 2000.

[26] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. 89:114–141, 2003.

[27] R. Collins, R. Gross, and J. Shi. Silhouette-based human identification from body shape and gait. In *Proc. of FGR*, pages 351–366, 2002.

[28] T. F. Cootes. Statistical models of appearance for computer vision. Technical report, University of Manchester, 2004.

[29] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. of ECCV*, volume 2, pages 484 – 498, 1998.

[30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and applications. *CVIU*, 61(1):38–59, 1995.

[31] T. Cox and M. Cox. *Multidimentional scaling*. Chapman & Hall, 1994.

[32] David Cunado, Mark S. Nixon, and John Carter. Automatic extraction and description of human gait models for recognition purposes. *CVIU*, 90:1–41, 2003.

[33] T. Darrell and A. Pentland. Space-time gesture. In *Proc. of CVPR*, pages 335–340, 1993.

[34] James W. Davis and Hui Gao. An expressive three-mode principal components model for gender recognition. *Journal of Vision*, 4(5):362–377, 2004.

[35] A. Elgammal, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *IEEE Proceedings*, 90(7):1151–1163, 2002.

[36] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. CVPR*, volume 2, pages 681–688, 2004.

[37] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proc. CVPR*, volume 1, pages 478–485, 2004.

[38] R. Fablet and M. J. Black. Automatic detection and tracking of human motion with a view-based representation. In *Proc. ECCV 2002, LNCS 2350*, pages 476–491, 2002.

[39] Brendan J. Frey and Nebojsa Jojic. Learning graphical models of images, videos and their spatial transformation. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence - San Francisco, CA*, pages 184 – 191, 2000.

[40] D. M. Gavrila. The visual analysis of human movement: a survey. *Comput. Vis. Image Underst.*, 73(1):82–98, 1999.

[41] D.M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Proc. of ICPR*, pages 439–444, 1998.

[42] D.M Gavrila and L.S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proc. of CVPR*, pages 73–80, 1996.

[43] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. ‘Dynamism of a dog on a leash’ or behavior classification by eigen-decomposition of periodic motions. In *Proceedings of the ECCV’02*, pages 461–475, Copenhagen, May 2002. Springer-Verlag, LNCS 2350.

[44] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison Wesley, 1992.

[45] Kristen Grauman, Gregory Shakhnarovich, and Trevor Darrell. Inferring 3d structure with a statistical image-based shape model. In *Proc. of ICCV*, page 641, Washington, DC, USA, 2003. IEEE Computer Society.

[46] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, 2nd edition, 1997.

[47] U. Grenander. *General Pattern Theory*. Oxford University Press, 1993.

[48] R. Gross and J. Shi. The cmu motion of body (mobo) database. Technical Report TR-01-18, Carnegie Mellon University, 2001.

[49] M. Prantl H. Borotschnig, L. Paletta and A. Pinz. Active object recognition in parametric eigenspace. In *Proc. of British Machine Vision Conference*, pages 629–638, 1998.

[50] Gregory D. Hager and Peter N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. PAMI*, 20(10), 1998.

[51] Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Sch&#246;lkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of ICML*, page 47, New York, NY, USA, 2004. ACM Press.

[52] James B. Hayfron-Aquah, Mark S. Nixon, and John N. Carter. Automatic gait recognition by symmetry analysis. *Pattern Recognition Letters*, 24:2175–2183, 2003.

[53] Q. He and C. Debrunner. Individual recognition from periodic activity using hidden markov models. In *In IEEE Workshop on Human Motion*, pages 47–52, 2000.

[54] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[55] Leventon Howe and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Proc. NIPS*, 1999.

[56] P.S. Huang, C.J. Haris, and M.S. Nixon. Recogising humans by gait via parametric canonical space. *Artificial Intelligence in Engineering*, 13:359–366, 1999.

[57] Yu Huang and Thomas S. Huang. Model-based human body tracking. In *ICPR (1)*, pages 552–555, 2002.

[58] Michael Isard and Andrew Blake. Condensation–conditional density propagation for visual tracking. *Int.J.Computer Vision*, 29(1):5–28, 1998.

[59] Odest Chadwicke Jenkins and Maja J. Matarić. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion. *International Journal of Humanoid Robotics*, 1(2):237–288, Jun 2004.

[60] Amos Y. Johnson and Aaron F. Bobick. A multi-view method for gait recognition using static body parameters. In *Proc. AVBPA*, pages 301–311, June 2001.

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

[62] J.O'Rourke and Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. PAMI*, 2(6), 1980.

[63] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, 1996.

[64] I. A. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pages 81–87, Los Alamitos, California, U.S.A., 18–20 1996. IEEE Computer Society.

[65] Amit Kale, Aravind Sundaresan, A. N. Rajagopalan, Naresh P. Cuntoor, Amit K. Roy-Chowdhury, Volker Kruger, and Rama Chellappa. Identification of human using gait. *IEEE Trans. Image Processing*, 13(9):1163–1173, 2004.

[66] Takeo Kanade, Yingli Tian, and Jeffrey F. Cohn. Comprehensive database for facial expression analysis. In *Proc. of FGR*, pages 46–53, 2000.

[67] A. Kapteyn, H. Neudecker, and T. Wansbeek. An approach to n-model component analysis. *Psychometrika*, 51(2):269–275, 1986.

[68] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(321-332), 1988.

[69] D. G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. London Math. Soc.*, 16, 1984.

[70] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

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

[72] Lieven De Lathauwer, Bart de Moor, and Joos Vandewalle. A multilinear singular value decomposiiton. *SIAM Journal On Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[73] 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*, 21(4):1324–1342, 2000.

[74] N. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In *NIPS*, 2003.

[75] N. D. Lawrence. Gaussian process models for visualisation of high dimensional data. In *In Proc. NIPS*, 2004.

[76] Chan-Su Lee and Ahmed Elgammal. Gait style and gait content: Bilinear model for gait recogntion using gait re-sampling. In *Proc. of FGR*, pages 147–152, 2004.

[77] Chan-Su Lee and Ahmed Elgammal. Facial expression analysis using nonlinear decomposable generative models. In *AMFG*, pages 17–31, 2005.

[78] Chan-Su Lee and Ahmed Elgammal. Homeomorphic manifold analysis: Learning decomposable generative models for human motion analysis. In *Workshop on Dynamical Vision*, 2005.

[79] Chan-Su Lee and Ahmed M. Elgammal. Simultaneous inference of view and body pose using torus manifolds. In *ICPR (3)*, pages 489–494, 2006.

[80] L. Lee, G. Dalley, and K. Tieu. Learning pedestrian models for silhouette refinement. In *Proc. of ICCV*, pages 663–670, 2003.

[81] M.E. Leventon, W. E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. CVPR*, pages 316–323, 2000.

[82] A. Levin and A. Shashua. Principal component analysis over continuous subspaces and intersection of half-spaces. In *Proc. of ECCV*, pages 635–650, 2002.

[83] Ying li Tian, Takeo Kanade, and Jeffrey F. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. PAMI*, 23(2), 2001.

[84] Jongwoo Lim and Ming-Hsuan Yang. A direct method for modeling non-rigid motion with thin plate spline. In *Proc. of CVPR*, volume 1, pages 1196–1202, 2005.

[85] James J. Little and Jeffrey E. Boyd. Recognizing people by their gait: The shape of motion. *Videre: Journal of Computer Vision Research*, 1(2), 1998.

[86] Zongyi Liu and Sudeep Sarkar. Simplest representation yet for gait recognition: Averaged silhouette. In *Proc. ICPR*, pages 211–214, 2004.

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

[88] David H. Marimont and Brian A. Wandell. Linear models of surface and illuminant spectra. *Journal of Optical Society of America*, 9(11):1905–1913, 1992.

[89] Iain Matthews and Simon Baker. Active appearance models revisited. 60(2):135–164, 2004.

[90] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.*, 81(3):231–268, 2001.

[91] Kooksang Moon and Vladimir Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *CVPR (1)*, pages 198–205, 2006.

[92] Vlad I. Morariu and Octavia I. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *CVPR (1)*, pages 545–552, 2006.

[93] Vlad I. Morariu and Octavia I. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 545–552, Washington, DC, USA, 2006. IEEE Computer Society.

[94] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, pages 666–680, 2002.

[95] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1):5–24, 1995.

[96] Hiroshi Murase and Rie Sakai. Moving object recognition in eigenspace representation: gait analysis and lip reading. *Pattern Recognition Letters*, 17:155–162, 1996.

[97] Kevin Murphy and Stuart Russell. *Sequential Monte Carlo Methods in Practice*, chapter 24 Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks, pages 499–515. 2001.

[98] 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*, pages 2–9, 2000.

[99] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[100] Nikos Paragios and Rachid Deriche. Geodesic active regions for motion estimation and tracking. In *ICCV (1)*, pages 688–694, 1999.

[101] P. Johathon Phillips, Sudeep Sarkar, Isidro Robledo, Patrick Grother, and Kevin Bowyer. Baseline results for the challenge problem of human id using gait analysis. In *Proc. of FGR*, pages 137–142, 2002.

[102] Rosalind W. Picard. Affective computing: Challenges. *Int. Journal of Human-Computer Studies*, 59(1–2):55–64, 2003.

[103] Tomaso Poggio and Fredrico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[104] Ali Rahimi, Ben Recht, and Trevor Darrell. Learning appearance manifolds from video. In *Proc. of CVPR*, volume 1, pages 868–875, Washington, DC, USA, 2005. IEEE Computer Society.

[105] James M. Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *ECCV (2)*, pages 35–46, 1994.

[106] James M. Rehg and Takeo Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617, 1995.

[107] K. Rohr. Towards model-based recognition of human movements in image sequence. *CVGIP*, 59(1):94–115, 1994.

[108] K. Rohr, H. S. Stiehl, T. M. Buzug, J. Weese, and M. H. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Trans. on Medical Imaging*, 20(6):526–534, 2001.

[109] R. Rosales, V. Athitsos, and S. Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Proc. ICCV*, pages 378–387, 2001.

[110] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *Workshop on Human Motion*, pages 19–24, 2000.

[111] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linar embedding. *Science*, 290(5500):2323–2326, 2000.

[112] Sudeep Sarkar, P. Jonathon Phillips, Zongyi Liu, Isidro Robledo Vega, Patrick Grother, and Kevin W. Bowyer. The humanid gait challenge problem: Data sets, performance, and analysis. *IEEE Trans. PAMI*, 27(2):162–177, 2005.

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

[114] H Sebastian Seung and Daniel D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.

[115] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face recognition from long-term observations. In *Proc. ECCV 2002, LNCS 2352*, pages 851–865, 2002.

[116] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. of ICCV*, pages 750–759, 2003.

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

[118] J. D. Shutler, M. G. Grant, M. S. Nixon, and J. N. Carter. On a large sequence-based human gait database. In *Proc. Int. Conf. on Recent Advances in Soft Computing*, pages 66–71, 2002.

[119] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.

[120] Leonid Sigal and Michael J. Black. Humaneva: Cynchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[121] Terence Sim and Sheng Zhang. Exploring face space. In *Workshop Proceedings FPIV*, 2004.

[122] Cristian Sminchisescu and Allan Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *Proceedings of ICML*, pages 96–103. ACM Press, 2004.

[123] Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris N. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR (1)*, pages 390–397, 2005.

[124] S. Soatto and A. Yezzi. Deformation: Defrormation motion, shape average and the joint registeration and segmentation of images. In *Proc. ECCV 2002, LNCS 2352*, pages 33–47, 2002.

[125] Mikkel B. Stegmann. Analysis and segmentation of face images using point annotations and linear subspace techniques. Technical Report TMM-REF-2002-22, Technical University of Denmark, 2002.

[126] J. Tenenbaum. Mapping a manifold of perceptual observations. In *Proc. of NIPS*, volume 10, pages 682–688, 1998.

[127] J.B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319 – 2323, 2000.

[128] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with biliear models. *Neural Computation*, 12:1247–1283, 2000.

[129] Tai-Peng Tian, Rui Li, and Stan Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 50, Washington, DC, USA, 2005. IEEE Computer Society.

[130] David Tolliver and Robert T. Collins. Gait shape estimation for identification. In *Proc. AVBPA*, pages 734–742, 2003.

[131] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *ICCV*, pages 50–59, 2001.

[132] Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[133] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[134] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, Washington, DC, USA, 2006. IEEE Computer Society.

[135] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. In *ICCV*, pages 403–410, 2005.

[136] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 403–410, Washington, DC, USA, 2005. IEEE Computer Society.

[137] M. Alex O. Vasilescu. Human motion signatures: Analysis, synthesis, recogntion. In *Proc. of ICPR*, volume 3, pages 456–460, 2002.

[138] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. of ECCV*, pages 447–460, 2002.

[139] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proc. of CVPR*, 2003.

[140] Hongcheng Wang and Narendra Ahuja. Facial expression decomposition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 958–965, 2003.

[141] Jack Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *NIPS*, 2005.

[142] Liang Wang, Tieniu Tan, Huazhang Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE Trans. PAMI*, 25(12):1505–1518, 2003.

[143] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. In *CVPR*, volume 2, pages 227–233, 2003.

[144] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and Peisen Huang. High resolution tracking of non-rigid 3d motion of densely sampled data using harmonic maps. In *ICCV'05*, pages I: 388–395, 2005.

[145] K. W. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of IEEE CVPR*, volume 2, pages 988–995, 2004.

[146] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, 1997.

[147] A. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neurosci*, 3(1):59–70, 1991.

[148] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *Proc. of FGR*, pages 336–341, 1998.

Vita

Chan-Su Lee

1990-95 Attended Yonsei University, Seoul, Korea. Majored in Electronics Engineering.

1995 B.A., Yonsei University.

1995-97 Graduate work in Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Taejon, Korea.

1997 M.S. in Electrical Engineering, KAIST.

1997-01 Engineering Staff at ETRI(Electronics and Telecommunications Research Institute), Taejon, Korea.

2001 Teaching Assistant, Rutgers, The State University of New Jersey, Piscataway, NJ.

2001-02 Programmer, Lab Assistant, Rutgers, The State University of New Jersey, Piscataway, NJ.

2002-07 Graduate Assistant, Rutgers, The State University of New Jersey, Piscataway, NJ.

2007 Ph.D. in Computer Science Rutgers, The State University of New Jersey, Piscataway, NJ.

Publications

- **Chan-Su Lee** and Ahmed Elgammal, ”Gait Style and Gait Content: Bilinear Model for Gait Recognition Using Gait-Resampling”, in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pp.147 - 152, 2004

- Ahmed Elgammal and **Chan-Su Lee**, ”Separating Style and Content on a Nonlinear Manifold”, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 478-485, 2004

- Ahmed Elgammal and **Chan-Su Lee**, ”Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning”, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 681-688, 2004

- Ahmed Elgammal and **Chan-Su Lee**, ”Separating Style and Content on a Nonlinear Manifold”, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 478-485, 2004

- Yang Wang, Xiaolei Huang, **Chan-Su Lee**, Song Zhang, Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang, ”High Resolution Acquisition, Learning and Transfer of Dynamic 3D Facial Expressions”, *Computer Graphics Forum*, vol. 23, no. 3, pp.677-686, 2004

- **Chan-Su Lee** and Ahmed Elgammal, ”Towards Scalable View-Invariant Gait Recognition: Multilinear Analysis for Gait”, in *Proc. of Audio- and Video-based Biometric Person Authentication(AVBPA) 2005, LNCS 3546*, pp. 395-405, 2005

- **Chan-Su Lee** and Ahmed Elgammal, ”Style Adaptive Bayesian Tracking Using Explicit Manifold Learning”, in *Proc. of British Machine Vision Conference* (BMVC), pp.739-748, 2005

- **Chan-Su Lee** and Ahmed Elgammal, ”Facial Expression Analysis using Nonlinear Decomposable Generative Models”, in *Proc. of IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG), LNCS3725*, pp.17-31, 2005

- **Chan-Su Lee** and Ahmed Elgammal, ”Gait Tracking and Recognition Using Person-Dependent Dynamic Shape Model” , in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pp.553-559, 2006

- **Chan-Su Lee**, Ahmed Elgammal and Dimitris Metaxas, ”Synthesis and Control of High Resolution Facial Expressions for Visual Interactions”, in *Proc. of IEEE International Conference on Multimedia & Expo (ICME)*, pp.64-67,2006

- **Chan-Su Lee** and Ahmed Elgammal, ”Carrying Object Detection Using Pose Preserving Dynamic Shape Model”, in Proc. of *International Conference of Articulated Motion and Deformable Objects (AMDO),LNCS4069*, pp.315-325, 2006

- **Chan-Su Lee** and Ahmed Elgammal, ”Human Motion Synthesis by Motion Manifold Learning and Motion Primitive Segmentation”, in *Proc. of International Conference of Articulated Motion and Deformable Objects (AMDO), LNCS4069*, pp.464-473, 2006

- **Chan-Su Lee** and Ahmed Elgammal, ”Nonlinear Shape and Appearance Models for Facial Expressions”, in Proc. of *IEEE International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 497-502, 2006

- **Chan-Su Lee** and Ahmed Elgammal, ”Simultaneous Inferring View and Body Pose Using Torus Manifolds”, in Proc. of *IEEE International Conference on Pattern Recognition (ICPR)*, vol.3, pp. 489-494,2006

- **Chan-Su Lee** and Ahmed Elgammal, ”Homeomorphic Manifold Analysis: Learning Decomposable Generative Models for Human Motion Analysis”, in *Proc. of IEEE International Workshop on Dynamical Vision (WDV), WDV 2005/2006, LNCS 4358*, pp. 100-114, 2007

- Ahmed Elgammal and **Chan-Su Lee**, ”Nonlinear Manifold Learning for Dynamic Shape and Dynamic Appearance”, *Computer Vision and Image Understanding (CVIU).*, vol. 106, no.1, pp.31-46, 2007