Dual Gait Generative Models for Human Motion Estimation From a Single Camera

Xin Zhang, *Student Member, IEEE*, and Guoliang Fan, *Senior Member, IEEE*

***Abstract*—This paper presents a general gait representation framework for video-based human motion estimation. Specifically, we want to estimate the kinematics of an unknown gait from image sequences taken by a single camera. This approach involves two generative models, called the kinematic gait generative model (KGGM) and the visual gait generative model (VGGM), which represent the kinematics and appearances of a gait by a few latent variables, respectively. The concept of gait manifold is proposed to capture the gait variability among different individuals by which KGGM and VGGM can be integrated together, so that a new gait with unknown kinematics can be inferred from gait appearances via KGGM and VGGM. Moreover, a new particle-filtering algorithm is proposed for dynamic gait estimation, which is embedded with a segmental jump–diffusion Markov Chain Monte Carlo scheme to accommodate the gait variability in a long observed sequence. The proposed algorithm is trained from the Carnegie Mellon University (CMU) Mocap data and tested on the Brown University HumanEva data with promising results.**

***Index Terms*—Gait appearances, gait kinematics, generative models, human motion estimation, manifold learning, Markov chain Monte Carlo (MCMC), particle filtering, tensor analysis.**

I. Introduction

VIDEO-BASED human motion analysis has recently received great interest due to its wide applications. On the one hand, it is a challenging topic due to the variability and nonlinearity of human motion as well as the uncertainty and ambiguity of visual observations. On the other hand, this topic has been advanced by recent progress in the fields of computer vision, artificial intelligence, machine learning, and image processing. In this paper, we are interested in the estimation of human body configurations from image sequences taken by a single collaborated camera. Specifically, we focus on the motion of walking (i.e., gait) that is useful for biometrics and many biomechanical modeling applications. Particularly, we define two terms about a gait: *gait kinematics* and *gait appearances*. The former one is represented by a sequence of Euler angles or 3-D positions of body joints, and the latter one is a sequence of human silhouettes extracted from an image sequence. Our goal is to estimate gait kinematics from gait appearances via explicit gait modeling in both kinematic and visual spaces. This paper addresses several fundamental issues pertaining to the emerging markerless motion capture technology [1].

We have two hypotheses in this paper. The first one is that we could span a nonlinear low-dimensional space to represent a variety of human gait motions (in terms of kinematics or appearances) by learning from a set of representative (training) gaits. The second one is that a new gait with unknown kinematics or appearances can be synthesized from the training gaits in this space. Particularly, we call this space *gait manifold*. It is worth noting that the term of gait manifold used here has been upgraded from its original meaning in some previous works, e.g., [2] and [3], where the gait manifold is referred to the low-dimensional intrinsic structure *among different poses* (either by their kinematics or appearances) *from a single gait*. Here, the gait manifold is used to represent the kinematic or visual variability *among different individuals*.

To address these two hypotheses, our paper involves three major components. First, we develop a general gait representation framework that involves dual gait generative models, i.e., the *kinematic gait generative model* (KGGM) and the *visual gait generative model* (VGGM). KGGM represents the kinematics of a gait by two variables, i.e., *gait* and *pose*, and VGGM characterizes the appearances of a gait by four variables, i.e., *view, shape, gait*, and *pose*. KGGM and VGGM are *temporally synchronized* by sharing the same pose variable. Second, we span a 1-D continuous gait manifold in each of the two generative models to capture the gait variability among different individuals, so that KGGM and VGGM can be *semantically integrated* by sharing the same gait manifold during training. This allows us to infer the kinematics of a new gait from its appearances. Third, considering the segmental variability of the gait variable in a long sequence, we develop an effective particle-filtering-based inference algorithm that is embedded with a segmental jump–diffusion Markov chain Monte Carlo (SJD-MCMC) scheme to support dynamic gait estimation. Two human motion databases were involved in our experiment, i.e., the Carnegie Mellon University Mocap [4] and the Brown HumanEva [5], which are used for algorithm training and testing, respectively. Both databases have been widely used by computer vision researchers who are interested in human motion analysis and pose estimation.

The remainder of this paper is organized as follows. After reviewing some related works in Section II, we present an overview of our approach in Section III. In Section IV, we discuss the learning of KGGM and VGGM that are the cornerstones of this paper. The concept of gait manifold is introduced

Manuscript received March 6, 2009; revised September 23, 2009; accepted January 21, 2010. Date of publication April 19, 2010; date of current version July 16, 2010. This work was presented in part at the International Conference on Pattern Recognition, Tampa, FL, December 2008. This work was supported in part by the National Science Foundation under Grant IIS-0347613 and in part by OHRS Award HR09-030 from the Oklahoma Center for the Advancement of Science and Technology. This paper was recommended by Associate Editor N. Rajpoot.

The authors are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74075 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2010.2044240

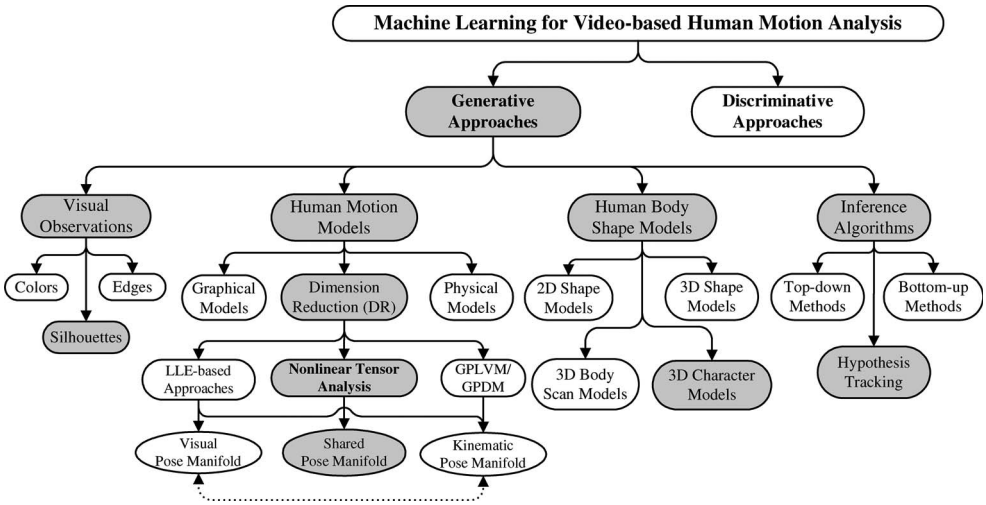


Fig. 1. Taxonomy of machine-learning-based human motion analysis where the shaded blocks indicate the choices in our paper.

in Section V. Section VI presents a new particle-filtering-based inference algorithm for dynamic gait estimation. The experimental results are shown in Section VII. The conclusion and future research are given in Section VIII.

II. Related Works

There have been a plethora of works on video-based human motion analysis. Previous surveys [6], [7] provide comprehensive reviews on this topic. Due to the nature of our research, we will present a brief review from the machine learning perspective where we mainly discuss *discriminative* and *generative* approaches. Discriminative approaches tend to directly learn a mapping function from visual observations to body configurations based on training data, where the key is to handle the ambiguity and uncertainty of visual data due to the pose/view variability. The methods in this category include the relevance vector machine [8], [9], probabilistic regression [10]–[13], the nearest neighbor [14], temporal chaining [15], and neural networks [16]. Generative approaches involve explicit models to explain the underlying visual/kinematic data via a few latent variables by which the motion/pose can be inferred. Generally speaking, discriminative approaches are more efficient and require less strict training, while generative ones are more effective to incorporate prior knowledge in the inference process. In this paper, we focus on generative approaches, because we want to develop a general gait modeling framework. Our discussion hereinafter will focus on the four major components in most generative approaches, as shown by Fig. 1.

A. Visual Observations

Visual observations are image descriptors extracted from image sequences for human body representation, including silhouettes, edges, colors, and their combinations [7]. Silhouettes [17]–[22] and edges [23]–[25] can be extracted efficiently from static background. Silhouettes are robust to color or texture variations of the human body but may be sensitive to the background noise or the shadow effect. Edge extraction may have some difficulties in cluttered background. The extended image descriptors, e.g., histogram of oriented gradients [26] and shape context [9], [27], can provide a rich and robust description of object shape. Color [28], [29] can also be used to represent individual body parts, but self-occlusion may impose some problems. The combination of multiple visual cues [30]–[32] proves to be useful. In our paper, we chose silhouette-based gait representation due to its robustness and simplicity.

B. Human Shape Models

Human shape models provide important shape priors to evaluate visual observations for pose/motion estimation, including 2-D/3-D shape models, 3-D body scan models, and 3-D computer models. Specifically, 2-D shape models [32], [33] use rectangles or ellipses to approximate each body part that can be adjusted by a couple of parameters but may be limited to deal with complex shapes and self-occlusion. Three-dimensional shape models [18], [25], [30], [31], [34] define the human body as an assembly of several rigid segments (i.e., cylinders or cuboids), each of which has a maximum of 3 degrees-of-freedom (DOFs). A 3-D body scan model from a laser scanner can be subject specific [35], [36] or general enough to handle various body shapes [27], [37], [38]. For example, the Shape Completion and Animation of PEople (SCAPE) is a data-driven human shape model that can span variation in both shape and pose [39]. Usually, using this model requires multiple cameras for accurate and robust estimation. Three-dimensional computer models [9], [17], [20], [22] are very cost effective and can be used for training data generation. Although each one is subject specific, multiple models can be used together to improve shape modeling even under a single camera [17]. Our paper involves five computer models.

C. Inference Algorithms

The inference process aims to find the optimal solution (including both motion and shape) that best explains visual observations. *Top–down* approaches [33], [34], [40] match the 2-D projection of a 3-D body part with visual observations, or called "analysis by synthesis." *Bottom–up* ones [25], [28], [29] search for all body parts and assemble them into a human body for motion/pose estimation, and they may not need

manual initialization. *Hypothesis tracking* can draw samples (or predictions) based on previous estimation and incorporate the temporal prior between poses or the spatial prior between parts for pose estimation. For example, particle-filtering-based tracking algorithms [30], [34] are often used for inference, which involve a dynamic model to predict a new pose [3] or the next part location [23]. Moreover, MCMC sampling can be embedded in the particle filter to further improve particle generation [41].

D. Human Motion Models

Motion models provide an important kinematic prior for generative models. *Graphic-model-based approaches* represent the spatial and temporal priors of body parts by learning from a set of labeled images [33] or motion capture data [25], [29]. *Physical-model-based approaches*[36], [42] incorporate various kinematic/dynamic constraints of body movements into the inference process. They may not need any training data, but a detailed physical model is hard to obtain, which may also impose some challenges for inference due to the high-dimensional nature of the model. For example, the methods in [23], [43], and [44] mainly focus on lower body motion. *Dimension-reduction (DR)-based methods* try to explore the low-dimensional intrinsic structure of human motion by learning from either kinematic or visual data that can be represented by a few latent variables and used for motion modeling.

Our paper is focused on DR-based motion modeling. There are two major DR approaches, i.e., *deterministic* and *probabilistic* ones. The former one includes Local Linear Embedding (LLE) [45] and Isomap [46] that can generate a latent space without providing a mapping function between the latent space and the data space. The latter one includes the Gaussian process latent variable model (GPLVM) [47] and its variants, such as Gaussian process dynamical models (GPDM) [48]–[50] and back-constrained GPLVM [51], [52], which can learn not only the latent space but also the mapping function. In most DR methods, a pose manifold is often involved that captures the pose variability of a particular motion, and it is usually a 1-D closed loop due to the cyclic nature of a gait.[1]

Single Pose Manifold: The pose manifold can be learned either from visual observations (e.g., silhouettes) by LLE [53], [54] or from kinematics data (i.e., motion capture data) by GPLVM or its variants [49], [52], [55]–[57]. The kinematic pose manifold provides an accurate dynamic model for part-based body tracking, and part-level shape models are needed (e.g., ellipsoids) to compute likelihoods that may encounter some difficulties due to the complexity of body parts. On the other hand, the visual pose manifold offers a direct way to generate visual hypotheses for likelihood computation, but it faces some challenges due to the problems of noise and one-to-many mapping as well as the view variability. For example, a view-dependent pose manifold was proposed in [53], where the view is treated as a discrete variable.

Dual Pose Manifolds: To take advantage of both kinematic and visual pose manifolds, dual pose manifolds were proposed for video-based pose/motion estimation [20]–[22]. Different DR methods were used to learn the kinematic and visual pose manifolds from the kinematic and visual data, respectively. In particular, a mapping function is needed between two pose manifolds by which the visual data can be associated with the kinematic data for motion estimation.

Shared Pose Manifold: In [17] and [58], a torus-shaped manifold is designed for joint view–pose modeling, which is shared in both kinematic and visual spaces. Two mapping functions are needed to map both kinematic and visual data onto the same torus manifold via radial basis functions (RBFs). Although this approach does not involve manifold learning, it provides promising pose tracking along with continuous view estimation. In [3], a kinematic pose manifold is first learned via LLE, which is also shared by the visual data using RBF-based mapping. Additionally, a continuous view manifold was proposed to support smooth view estimation, which plays an essential role during simultaneous pose/view estimation.

In most DR-based methods, the same subject is used for training and testing, and our paper aims at estimating a new gait from an unknown person. Specifically, our paper is inspired by the nonlinear tensor analysis approach proposed in [3] that combines manifold learning with multilinear analysis and provides a compact generative model to represent a series of visual observations (e.g., silhouettes) from one subject by multiple factors, including pose, view, and shape.

III. Research Overview

This section presents an overview of this paper, which briefly discusses three major technical issues, two gait generative models, the gait manifold, and dynamic gait estimation.

A. Dual Gait Generative Models

We propose KGGM and VGGM for gait representation in the kinematic and visual spaces, respectively. Specifically, KGGM represents gait kinematics by two latent variables, namely, *pose* and *gait*, where the pose variable defines a specific body configuration during a gait (or a gait phase) and the gait variable represents a specific gait motion. VGGM represents gait appearances by four latent variables, namely, *pose*, *gait*, *view*, and *shape*, where the pose and gait variables are similar to the ones in KGGM and the view and shape variables reflect the view angle and the appearance of the subject, respectively. Both KGGM and VGGM can be learned from a set of training data by extending the nonlinear tensor decomposition method proposed in [3]. The learning of KGGM uses a set of gait motion data (or gait kinematics) acquired by a motion capture system. The same set of motion data is also used to generate a set of gait animations by commercial software that are used for learning VGGM. KGGM and VGGM are fully *temporally synchronized* by sharing the same pose variable during model learning, and they are the cornerstone of this paper.

B. Gait Manifolds

In this paper, we advocate the concept of *gait manifold* that captures the gait variability among different individuals and could span a low-dimensional latent space to represent all

[1]There are several names in the literature for this concept, including the kinematic manifold [3], the gait manifold [2], or the pose latent space [20].

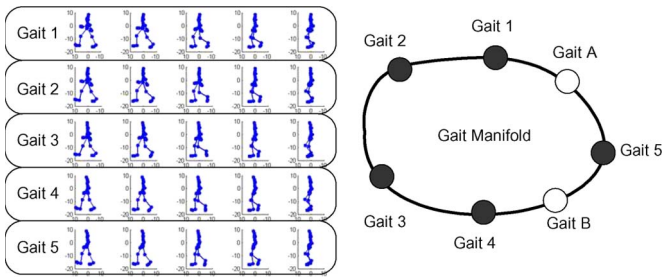


Fig. 2. Illustration of gait manifold used for gait synthesis.

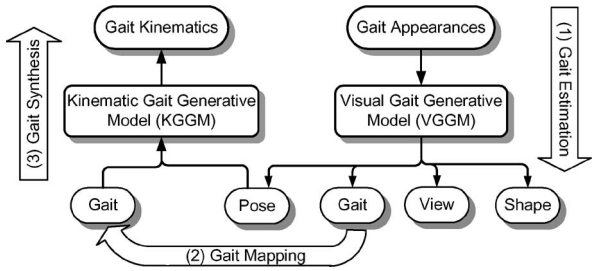


Fig. 3. Algorithm flow. (1) Gait estimation via VGGM. (2) Gait manifold mapping (VGGM → KGGM). (3) Gait synthesis via KGGM.

human gaits. This gait manifold can be learned from a set of representative training gaits, and it is defined at the gait variable of either KGGM or VGGM by which we can synthesize a new gait in terms of its kinematics or appearances.

As shown in Fig. 2, given the motion data of five training gaits (Gaits 1–5), we can represent them in a low-dimensional (e.g., 2-D) latent space where each gait is represented by a 2-D vector (or gait vector). By treating the five gait vectors as the anchor points, we can span a 1-D closed-loop gait manifold via curve fitting, where a new gait (Gait A or B) can be synthesized by nonlinear interpolation. We assume a 1-D nonlinear structure due to the simplicity of manifold generation, and the reason of using a closed loop is to ease the inference process.

Three questions need to be answered to develop such gait manifold: 1) how to represent all training gaits in a low-dimensional latent space as a set of gait vectors; 2) how to determine an *optimal* ordering relationship of all training gaits; and 3) how to connect them into a continuous closed loop that supports meaningful gait synthesis. Particularly, we propose a new manifold learning technique that can span a gait manifold in the tensor coefficient space of the gait variable in KGGM or VGGM. Correspondingly, two gait manifolds, namely, the *kinematic gait manifold* and the *visual gait manifold*, are obtained, which represent the variability of gait kinematics and that of gait appearances among different individuals, respectively. After ensuring that the two gait manifolds share the same topology (i.e., the ordering relationship of all training gaits along the manifold), we can learn a nonlinear mapping function to associate the two gait manifolds, by which KGGM and VGGM can be *semantically integrated*.

C. Inference for Gait Estimation

The key issue is how to estimate the underlying gait kinematics from the observed gait appearances via VGGM and KGGM. The generative model approach fits well in the Bayesian approach, which attempts to construct the posterior probability density function of the states based on all states and observations available. Here, the states to be estimated are the four latent variables of VGGM, and the observation is a series of gait appearances. We develop a particle-filtering-based inference algorithm. Specifically, we have one important observation for gait estimation in a long sequence. Although the gait variable is usually stable within each half cycle of a gait, it may vary near a particular pose, particularly when the subject is not walking straight or walking around a circle. Hence, we develop a new SJD-MCMC scheme that combines the ideas of segmental modeling and jump–diffusion MCMC and is incorporated into the particle filter for dynamic gait estimation.

The algorithm flow is shown in Fig. 3, where three steps are involved. The first is *gait estimation*, where an unknown gait is inferred from gait observations via VGGM by using a particle-filtering algorithm. The second is *gait manifold mapping*, which maps the estimated gait variable from VGGM to KGGM via a nonlinear manifold mapping. The third is *gait synthesis* by KGGM, which yields the estimated gait kinematics.

IV. Dual Gait Generative Models

The learning of KGGM and VGGM is essential in a DR process, where we extend the nonlinear tensor decomposition method proposed in [3], as shown in Fig. 4.

A. KGGM

Gait kinematics can be represented in different ways, like the 3-D positions of all joints or the angles between two adjacent joints. In order to reduce the effect of skeleton variability, gait kinematics are represented by a sequence of relative Euler angles between two adjacent joints. To learn KGGM, we need a universal pose manifold shared by different gaits, based on which we can develop a unified gait representation. However, the pose manifold varies from gait to gait. Inspired by the conceptual torus manifold proposed in [58], we define a circular-shaped conceptual manifold in a 2-D space to represent a general pose variation in one gait cycle. The learning of KGGM is shown in Fig. 4 (the left side).

Let $\mathcal{Z} = \{\mathbf{z}^{(i,q)} | i = 1, \ldots, N_g, q = 1, \ldots, N_p\}$ denote the set of N_g training gaits of N_p poses, where $\mathbf{z}^{(i,q)} \in \mathbb{R}^k$ encodes the k-dimensional kinematics for pose q of gait i. All poses are denoted by a set of 2-D coordinates, i.e., $\{\mathbf{p}_q \in \mathbb{R}^2, q = 1, \ldots, N_p\}$, uniformly sampled along the pose manifold. A nonlinear mapping function from $\mathbf{p}_q$ to $\mathbf{z}^{(i,q)}$ $(\mathbb{R}^2 \rightarrow \mathbb{R}^k)$ can be learned via a generalized RBF as

$$\mathbf{z}^{(i,q)} = \mathbf{B}^i \psi(\mathbf{p}_q) \tag{1}$$

where $\psi(\cdot)$ is a nonlinear kernel function defined as

$$\psi_L(\mathbf{p}_q) = \left[\phi\left(\mathbf{p}_q, \mathbf{c}_p^1\right), \ldots, \phi\left(p_q, \mathbf{c}_p^L\right)\right] \tag{2}$$

where $\phi(\cdot, \cdot)$ is an RBF (here we use Gaussian) and $\{\mathbf{c}_p^l | l = 1, \ldots, L\}$ denotes the kernel centers along the pose manifold. $\mathbf{B}^i$ represents a $k \times L$ linear mapping matrix that encodes the individuality of training gait i.

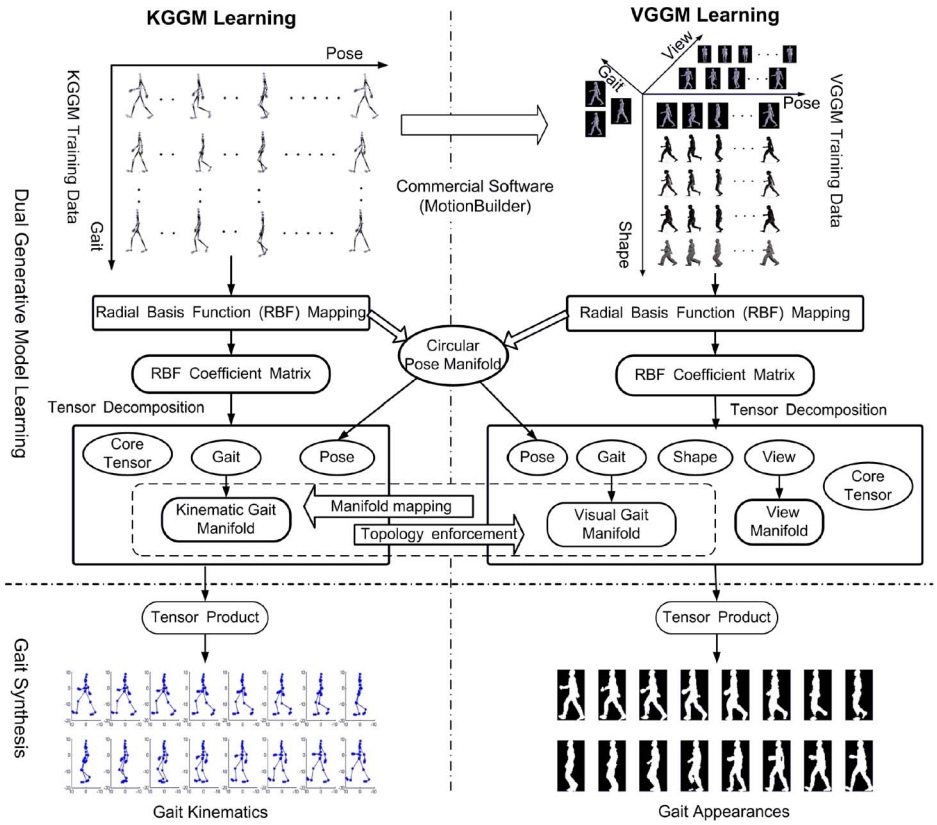


Fig. 4. Learning of KGGM and VGGM.

All gait-dependent mapping matrices $\{\mathbf{B}^i | i = 1, \ldots, N_g\}$ can be stacked as a tensor, and the higher order singular value decomposition [59] can be applied to decompose the tensor into two independent variables, i.e., the pose and gait. Then, the generative model is defined as

$$\mathbf{z}^{(i,q)} = \mathcal{A} \times_1 \boldsymbol{\kappa}^i \times_2 \psi(\mathbf{p}_q) \tag{3}$$

where $\mathcal{A}$ is called *core tensor* ($k \times N_g \times L$) governing the interaction between two variables, $\boldsymbol{\kappa}^i$ ($N_g \times 1$) represents gait i, and $\times_j$ denotes mode-j tensor product. Given a gait coefficient, this KGGM can synthesize the kinematics of an arbitrary pose.

B. VGGM

We use commercial 3-D animation software MotionBuilder to generate a set of gait animations, which involves multiple 3-D human models and the same set of gait motion data used for learning KGGM. Each human model can be driven by N_g gait motions to produce different animations, each of which can be recorded under different camera views. Specifically, we use a global feature-free representation to represent gait appearances that can be obtained by the signed distance transform of the body silhouette extracted from an image [2]. The learning of VGGM is similar to that of KGGM but with more factors involved, as shown in Fig. 4 (the right side).

Let $\mathcal{Y} = \{\mathbf{y}^{(k,j,i,q)} | k = 1, \ldots, N_v, j = 1, \ldots, N_s, i = 1, \ldots, N_g, q = 1, \ldots, N_p\}$ represent the set of training gait appearances, where $\mathbf{y}^{(k,j,i,q)} \in \mathbb{R}^d$ is the d-dimensional gait appearance of gait i, pose q, shape j, and view k; and N_v, N_s, N_g, and N_p are the numbers of views, shapes, gaits, and poses, respectively. By sharing the same pose manifold with KGGM, we can represent a gait appearance by four factors, i.e., the pose, gait, view, and shape, by assuming that they are independent as follows:

$$\mathbf{y}^{(k,j,i,q)} = \mathcal{C} \times_1 \mathbf{v}^k \times_2 \mathbf{s}^j \times_3 \boldsymbol{\nu}^i \times_4 \varphi(\mathbf{p}_q) \tag{4}$$

where $\mathcal{C}(d \times N_v \times N_s \times N_g \times N_p)$ is the fifth-order core tensor governing the interaction between four variables; $\varphi(\cdot)$ is a nonlinear kernel function similar to the one in (2); and $\mathbf{v}^k$, $\mathbf{s}^j$, $\boldsymbol{\nu}^i$, and $\mathbf{p}_q$ represent the view, shape, gait, and pose, respectively. The first two are unique for VGGM, and the latter two have a close relationship with their counterparts in KGGM. Similar to [3], we can learn a *view manifold* from $\{\mathbf{v}^k | k = 1, \ldots, N_v\}$. Given a gait coefficient, a view coefficient along the view manifold, and a shape coefficient, this VGGM can synthesize the gait appearance of an arbitrary pose.

C. Mapping Between KGGM and VGGM

Since KGGM and VGGM share the same pose manifold during the learning process, the pose variables in (3) and (4) are equivalent and identical. However, the gait variables, i.e., $\boldsymbol{\kappa}^i$ in (3) and $\boldsymbol{\nu}^i$ in (4), are quite different, since they represent the individuality of a training gait in the kinematic and visual spaces, respectively. The two generative models are not ready to be *integrated together* yet due to the incompatibility between $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$. In the following, we want to find a mapping relationship

between the two gait variables to bridge the gap between KGGM and VGGM.

V. Gait Manifolds

The primary goal of our paper is to use the training gaits to synthesize a new gait with unknown kinematics or appearances, where the concept of *gait manifold* plays an important role, by which a new gait is interpolated from training gaits via VGGM or KGGM. In the following, we discuss three related issues: 1) how to create two gait manifolds, i.e., one from KGGM and one from VGGM; 2) how to ensure the compatibility between the two gait manifolds that reflect the different natures of a gait; and 3) how to integrate KGGM and VGGM by establishing a mapping relationship between the two gait manifolds for gait estimation and synthesis.

A. Gait Manifold Generation

In either KGGM or VGGM, we have N_g gait coefficients ($\{\boldsymbol{\kappa}^i | i = 1, \ldots, N_g\}$ or $\{\boldsymbol{\nu}^i | i = 1, \ldots, N_g\}$) representing N_g training gaits. To derive a 1-D closed-loop gait manifold, we need to know the *order*, along which these gait coefficients can be connected. This order is referred to as the *manifold topology*. However, unlike the case of view manifold generation in [3], where the intrinsic order of multiple views is available, there does not exist any explicit ordering relationship among N_g training gaits. In other words, we do not know the topology of the gait manifold among the N_g training gaits. In this paper, we assert that the shortest path connecting all gait coefficients in the tensor coefficient space provides the *optimal* 1-D manifold topology, because the shortest path ensures the best local linearity and smoothness in the gait manifold that is important for nonlinear interpolation for gait synthesis. We tested this method in the view coefficient space and found that it produces the correct view order reflecting the orderly change of view angles. Therefore, we can learn two manifold topologies from KGGM and VGGM, respectively

$$\mathcal{T}_\kappa = \arg\min_{Q} \sum_{i=1}^{N_g} \mathcal{D}\left(\boldsymbol{\kappa}^{(q_i)}, \boldsymbol{\kappa}^{(q_{i+1})}\right) \tag{5}$$

$$\mathcal{T}_\nu = \arg\min_{Q} \sum_{i=1}^{N_g} \mathcal{D}\left(\boldsymbol{\nu}^{(q_i)}, \boldsymbol{\nu}^{(q_i+1)}\right) \tag{6}$$

where $Q = \{q_i \in [1, N_g] | i = 1, \ldots, N_g + 1, q_i \neq q_j \text{ for } i \neq j; q_1 = q_{N_g} + 1\}$ specifies an order to connect N_g training gaits in a closed loop; $\mathcal{T}_\kappa$ and $\mathcal{T}_\nu$ specify the *optimal* order of the shortest closed pathes in KGGM and VGGM, respectively; and $\mathcal{D}$ defines the Euclidean distance in the tensor coefficient space.[2] Correspondingly, the two continuous 1-D nonlinear gait manifolds are obtained as

$$\mathcal{M}_\kappa = \mathcal{S}\left(\boldsymbol{\kappa}^{\mathcal{T}_\kappa(i)} \middle| i = 1, \ldots, N_g\right) \tag{7}$$

$$\mathcal{M}_\nu = \mathcal{S}\left(\boldsymbol{\nu}^{\mathcal{T}_\nu(i)} \middle| i = 1, \ldots, N_g\right) \tag{8}$$

[2]Since all gait vectors are mutually orthogonal, we have to remove the last two dimensions of each gait vector for distance computation.

where $\mathcal{S}$ is the spline fitting function to connect the N_g training gait coefficients, i.e., $\boldsymbol{\kappa}^{\mathcal{T}_\kappa(i)}$ or $\boldsymbol{\nu}^{\mathcal{T}_\nu(i)}$, in the tensor coefficient space defined by KGGM in (3) or VGGM in (4), respectively. $\mathcal{M}_\kappa$ and $\mathcal{M}_\nu$ are called the *kinematic* and *visual gait manifolds* that represent the variability among all training gaits in terms of their kinematics and appearances, respectively.

B. Manifold Topology Enforcement

Essentially, a gait manifold will determine how nonlinear interpolation is performed for gait synthesis. Due to their different natures, the two gait manifolds, namely, $\mathcal{M}_\kappa$ and $\mathcal{M}_\nu$, usually have different topologies regarding the order of the training gaits along the gait manifold. As mentioned before, we are interested in estimating the unknown gait kinematics from the observed gait appearances. Thus, the kinematic gait manifold, i.e., $\mathcal{M}_\kappa$, is more critical, which directly affects the performance of gait synthesis in terms of the capability of interpolating new gait kinematics. Therefore, we develop a *manifold topology enforcement* scheme to ensure the visual gait manifold to share the same topology with $\mathcal{M}_\kappa$. Then, $\mathcal{M}_\nu^*$ denotes the visual gait manifold after topology enforcement, represented as

$$\mathcal{M}_\nu^* = \mathcal{S}\left(\boldsymbol{\nu}^{\mathcal{T}_\kappa(i)} \middle| i = 1, \ldots, N_g\right). \tag{9}$$

Now, $\mathcal{M}_\kappa$ and $\mathcal{M}_\nu^*$ share the same topology and are ready for nonlinear manifold mapping.

C. Mapping Between the Two Gait Manifolds

Basically, $\mathcal{M}_\kappa$ and $\mathcal{M}_\nu^*$ define the gait variable in KGGM and that in VGGM, respectively. The mapping between two manifolds will naturally lead to the integration of KGGM and VGGM via their gait variables. Specifically, we develop an RBF-based mapping between the two gait manifolds as

$$\boldsymbol{\kappa}^i = \mathcal{F}(\boldsymbol{\nu}^i) = \sum_{j=1}^{J} \omega_j \zeta\left(\boldsymbol{\nu}^i - \mathbf{c}_\nu^j\right) \tag{10}$$

where $\mathcal{F}(\cdot)$ maps $\boldsymbol{\nu}^i \in \mathcal{M}_\nu^*$ to $\boldsymbol{\kappa}^i \in \mathcal{M}_\kappa$ for each training gait; $\{\mathbf{c}_\nu^j | j = 1, \ldots, J\}$ denotes the kernel centers along $\mathcal{M}_\nu^*$; and $\zeta(\cdot)$ is a Gaussian function. Given a new gait that is between two training gaits along $\boldsymbol{\nu}' \in \mathcal{M}_\nu^*$, we can map it to $\mathcal{M}_\kappa$ as

$$\boldsymbol{\kappa}' = \arg\min_{\boldsymbol{\kappa}} \mathcal{D}\left(\mathcal{F}(\boldsymbol{\nu}'), \boldsymbol{\kappa} \,|\, \boldsymbol{\kappa} \in \mathcal{M}_\kappa\right) \tag{11}$$

where $\boldsymbol{\kappa}'$ is the corresponding gait variable in KGGM.

Fig. 3 shows how KGGM and VGGM along with the two gait manifolds are used for motion estimation. The gait variable is first estimated along $\mathcal{M}_\nu^*$ via VGGM, where both observed and synthesized gait appearances are involved in the gait estimation process. Then, the estimated gait variable is mapped to the one in $\mathcal{M}_\kappa$ via $\mathcal{F}(\cdot)$, and the underlying gait kinematics can be synthesized by KGGM according to (3).

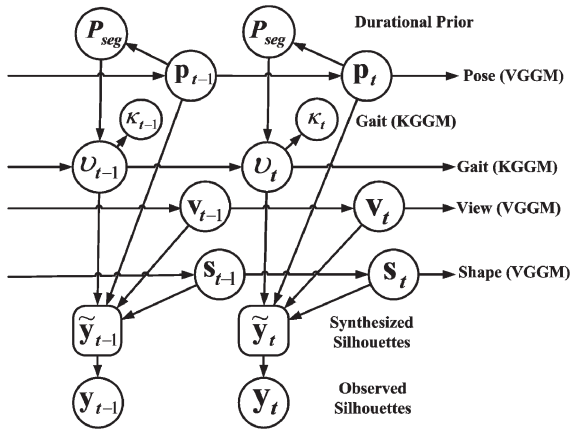


Fig. 5. Graphic model for gait tracking and estimation.

VI. Inference Algorithm

This section discusses two inference issues: 1) how to formulate gait estimation as an inference problem that involves multiple latent variables in VGGM and 2) how to cope with the dynamic nature of the gait variable in a long sequence.

A. Graphical Models

VGGM specifies four latent variables, namely, pose $\mathbf{p}_t$, view $\mathbf{v}_t$, shape $\mathbf{s}_t$, and gait $\boldsymbol{\nu}_t$, all of which have to be estimated. We employ a graphical model to integrate all related variables along with their conditional dependences, as shown in Fig. 5. According to Bayes' rule, we can recursively estimate the posterior for time step t, and latent variables are estimated via maximum *a posteriori* probability (MAP) as

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{y}_{t-1}) p(\mathbf{y}_t \mid \mathbf{x}_t) \tag{12}$$

where $\mathbf{x}_t = [\mathbf{p}_t, \mathbf{v}_t, \mathbf{s}_t, \boldsymbol{\nu}_t]$ encapsulates the four latent variables; $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ specifies the prediction based on previous observation $\mathbf{y}_{t-1}$, and $p(\mathbf{y}_t \mid \mathbf{x}_t)$ defines the observation model that involves the comparison between the observed gait appearance $\mathbf{y}_t$ and the one synthesized by VGGM, given four hypothesized latent variables, as defined in the following:

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = p(\mathbf{y}_t \mid \mathbf{p}_t, \mathbf{v}_t, \mathbf{s}_t, \boldsymbol{\nu}_t) \propto \exp -\frac{\|\mathbf{y}_t - \tilde{\mathbf{y}}_t\|^2}{2\sigma^2} \tag{13}$$

where $\tilde{\mathbf{y}}_t$ is synthesized by VGGM according to (4), $\|\cdot\|^2$ denotes the mean square error, and σ_2 controls the sensitivity of observation evaluation.

Since the four variables are independent, we can approximate $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ in (12) by the product of the prior distribution of each latent variable, given previous state estimation, i.e.,

$$p(\mathbf{x}_t \mid \mathbf{y}_{t-1}) \approx p(\mathbf{p}_t \mid \mathbf{x}_{t-1}) p(\mathbf{v}_t \mid \mathbf{x}_{t-1}) p(\mathbf{s}_t \mid \mathbf{x}_{t-1}) p(\nu_t \mid \mathbf{x}_{t-1}) \tag{14}$$

where four dynamic models are involved for four latent variables. Specifically, three of them are constrained by their 1-D nonlinear manifold, i.e., the pose manifold for $\mathbf{p}_t$, the view manifold for $\mathbf{v}_t$, and the visual gait manifold (i.e., $\mathcal{M}^*_\nu$) for $\boldsymbol{\nu}_t$. The shape variable $\mathbf{s}_t = [w_t^1, \ldots, w_t^{N_s} \mid \sum_{j=1}^{N_s} w_t^j = 1]$ represents the linear combination coefficients of N_s prototype shapes specified by $\{\mathbf{s}^j \mid j = 1, \ldots, N_s\}$ given in (4). Therefore, we use a constant speed dynamic model for $\mathbf{p}_t$ defined along the circular-shaped pose manifold, and a random walk to propagate the view samples along the view manifold. We also use a random walk to sample the shape variable in the tensor coefficient space. Regarding $\boldsymbol{\nu}$, we need a special dynamics to sample it along the gait manifold due to its segmental variability that will be discussed shortly.

For simplicity, we can sequentially estimate four latent variables one by one in the order of their robustness, i.e., pose → view → shape → gait → pose → view → shape. In other words, (12) is decomposed into four steps where the four latent variables are estimated individually and sequentially. The pseudocode of the complete inference algorithm is illustrated in Algorithm 1 in the following. For each variable, we resort to the MCMC sampling approach to rejuvenate the sample distribution for state estimation in each time step. The inference algorithm is not sensitive to the initialization that only needs a rough estimation of pose and view parameters to reduce the ambiguity introduced by silhouettes from a single camera.

Algorithm 1 Inference Algorithm

1: Given observations $\mathbf{y}_t$ and previous state estimation $\mathbf{x}_{t-1}$;
2: Predict pose $\mathbf{p}'_t$, view $\mathbf{v}'_t$, shape $\mathbf{s}'_t$, and gait $\boldsymbol{\nu}'_t$ according to their own dynamic models;
3: Update pose $\mathbf{p}''_t$ using MCMC sampling, given $\mathbf{v}'_t$, $\mathbf{s}'_t$, and $\boldsymbol{\nu}'_t$;
4: Update view $\mathbf{v}''_t$ using MCMC sampling, given $\mathbf{p}''_t$, $\mathbf{s}'_t$, and $\boldsymbol{\nu}'_t$;
5: Update shape $\mathbf{s}''_t$ using MCMC sampling, given $\mathbf{p}''_t$, $\mathbf{v}''_t$, and $\boldsymbol{\nu}'_t$;
6: Estimate gait $\hat{\boldsymbol{\nu}}_t$, as discussed in Section VI-B;
7: Refine pose $\hat{\mathbf{p}}_t$ using MCMC sampling, given $\mathbf{v}''_t$, $\mathbf{s}''_t$, and $\hat{\boldsymbol{\nu}}_t$;
8: Refine view $\hat{\mathbf{v}}_t$ using MCMC sampling, given $\hat{\mathbf{p}}_t$, $\mathbf{s}''_t$, and $\hat{\boldsymbol{\nu}}_t$;
9: Refine shape $\hat{\mathbf{s}}_t$ using MCMC sampling, given $\hat{\mathbf{p}}_t$, $\hat{\mathbf{v}}_t$, and $\hat{\boldsymbol{\nu}}_t$.

(Note: $\mathbf{p}'_t$, $\mathbf{p}''_t$, and $\hat{\mathbf{p}}_t$ represent the *predicted*, *updated*, and *refined* pose variables, respectively. The same notation applies to other variables.)

B. Dynamic Gait Estimation

In practice, we observed that, in a long gait sequence, the gait variable may exhibit significant variations, particularly when that person is not walking straight or is walking around a circle. Usually, the gait variable is dominated by one value within each half cycle, and it may *jump* to another value (along the gait manifold) in the next half cycle. We believe that it is because the 1-D gait manifold used here is a simplified representation of the unknown gait space that is very likely to be a high-dimensional one, and the continuity in that space is not well preserved in this 1-D space. Moreover, we also observed that

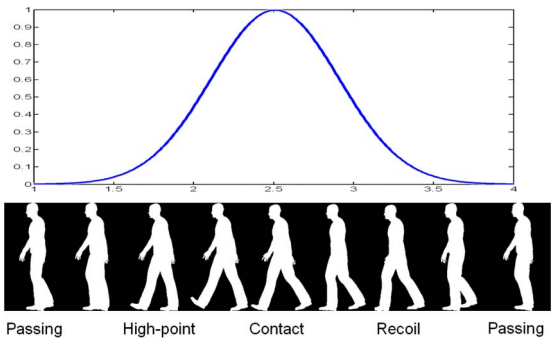


Fig. 6. Segmental prior model, where the pose is defined according to www.idleworm.com/how/anm/02w/walk1.shtml.

the jump usually occurs around the *contact* pose when both feet are on the ground and a new half cycle starts. To accommodate these two factors, we propose a new SJD-MCMC inference scheme that is embedded in the particle filter for dynamic gait estimation. Specifically, *jump* enables the sample generation to traverse along the gait manifold and to explore globally, while *diffusion* draws samples intensively in a local area of the gait manifold.

Since the gait manifold is a 1-D closed loop in the high-dimensional tensor coefficient space, we represent it by a 2-D circular gait manifold in order to facilitate the inference process. Then, the gait is denoted by an angular variable $\boldsymbol{\nu} \in [0, 2\pi)$, and we can define a one-to-one relationship between this angular gait manifold and the original one defined in the tensor coefficient space, i.e., $\boldsymbol{\nu} = \mathfrak{f}(\nu)$. For simplicity, we will use them exchangeably in the following discussion.

1) Segmental Modeling: The idea of segmental modeling allows us to control the switch between two kinds of dynamics, namely, jump and diffusion, in a probabilistic way. In particular, we define a probabilistic model of duration prior (as shown in Fig. 6), which is a function of pose, as follows:

$$P_{\text{seg}} = \beta \exp \frac{-\tau_{\mathbf{p}}^2}{2\sigma_{\mathbf{p}}^2} \sim (0, 1] \tag{15}$$

where $\tau_{\mathbf{p}}$ is the arc distance between the current pose and the contact pose along the pose manifold and $\sigma_{\mathbf{p}}^2$ controls the relative frequency of *jump*. This model indicates how likely a *jump* should be triggered, given current pose estimation.

2) Mode-Based Gait Estimation: Due to the hybrid nature of the gait variable, we will use the jump–diffusion MCMC to generate gait samples. The challenge is that the sampling space is continuous, which is different from traditional *jump–diffusion* applications where a mixed discrete continuous state space is involved [60]. Therefore, we propose the concept of *mode*. A mode is a local continuous model defined along the angular gait manifold, while a set of modes cover all possible gait values. Then, *jump* indicates switches among modes, and *diffusion* exploits within a mode. Gait estimation will involve several modes defined as

$$\mathbb{M}_{(1,\ldots,R)} = \left\{ \left(\mu_1, \delta_1^2\right), \ldots, \left(\mu_R, \delta_R^2\right) \right\} \tag{16}$$

where $\mathbb{M}_{(1,\ldots,R)}$ represents R modes, and each mode $\mathbb{M}_\tau$ is a Gaussian function with mean μ_τ and variance δ_τ^2 defined along the angular gait manifold. For an unknown test subject, we will estimate online these modes from which we can draw samples along the angular gait manifold that can be further mapped to the hypothesized gait coefficients in the tensor space via $\boldsymbol{\nu} = \mathfrak{f}(\nu)$ for VGGM-based gait synthesis defined in (4). Initially, we can define a fixed number of modes that are uniformly distributed along the angular gait manifold with equally large variances to cover all possible gait values. During inference, the mean and variance of some modes will be updated, as discussed in the following.

3) SJD-MCMC Inference: We use the Metropolis–Hasting algorithm as the inference framework that incorporates the following two kinds of dynamics.

1) **Jump**: At the ith MCMC iteration, the state vector is denoted $\mathbf{x}^{(i)}$ that includes the gait sample $(\nu^{(i)})$, and the gait mode is $(m^{(i)})$. We randomly choose a mode $m^* \in [1, R]$ with a probability $1/R$ and generate a new sample ν^* according to $\mathbb{M}_{m^*}$. The proposal distribution defined is independent with $\nu^{(i)}$ and $m^{(i)}$

$$q\left(\nu^*, m^* \,\middle|\, \nu^{(i)}, m^{(i)}\right) = N\left(\nu^*; \mu_{m^*}, \delta_{m^*}^2\right). \tag{17}$$

Hence, the acceptance ratio is

$$\alpha = \min\left\{1, \frac{p(\mathbf{x}^* \,|\, \mathbf{y}) N\left(\nu^{(i)}; \mu_{m^{(i)}}, \delta_{m^{(i)}}^2\right)}{p\left(\mathbf{x}^{(i)} \,|\, \mathbf{y}\right) N\left(\nu^*; \mu_{m^*}, \delta_{m^*}^2\right)}\right\} \tag{18}$$

where $\mathbf{x}^*$ is a new version of $\mathbf{x}^{(i)}$ by incorporating ν^* as the gait sample. $p(\mathbf{x}^* \,|\, \mathbf{y})$ is the posterior probability computed by the product of (13) and (14).

2) **Diffusion**: We randomly sample the gait variable ν^* with a proposal distribution as

$$q\left(\nu^* \,\middle|\, \nu^{(i)}\right) = N\left(\nu^*; \nu^{(i)}, \delta_d^2\right) \tag{19}$$

where δ_d^2 is the diffusion variance, which is decreased in a simulated annealing way. The acceptance ratio is

$$\begin{aligned}\alpha &= \min\left\{1, \frac{p(\mathbf{x}^* \,|\, \mathbf{y}) N\left(\nu^{(i)}; \nu^*, \delta_d^2\right)}{p\left(\mathbf{x}^{(i)} \,\middle|\, \mathbf{y}\right) N\left(\nu^*; \nu^{(i)}, \delta_d^2\right)}\right\} \\ &= \min\left\{1, \frac{p(\mathbf{x}^* \,|\, \mathbf{y})}{p\left(\mathbf{x}^{(i)} \,\middle|\, \mathbf{y}\right)}\right\}. \end{aligned} \tag{20}$$

The pseudocode of the SJD-MCMC scheme is presented in Algorithm 2 that is the core of our inference algorithm and embedded in each time step of the particle filter.

Algorithm 2 SJD-MCMC Inference

1: Initialization: Let the initial MCMC sample $\nu_t^{(0)}$ be the MAP-estimated gait variable from the previous time step, and keep the previous gait mode, i.e., $\nu_t^{(0)} = \hat{\nu}_{t-1}$ and $m_t^{(0)} = m_{t-1}$.
2: Compute the segmental probability P_{seg} using (15).
3: **for** $i = 1, \ldots, (B + MN)$ (N is the number of samples, B is the length of the burn-in period. and M is the length of the thinning interval) **do**

4: Randomly sample $\gamma \sim U[0,1]$.
5: **if** $P_{\text{seg}} \geq \gamma$ **then**
6: **Jump** Sample the mode variable $m^* \in [1, R]$ and the gait variable ν^* according to (17). Compute the acceptance ratio α using (18).
7: **else**
8: **Diffusion** Sample ν^* according to (19). Compute the acceptance α ratio using (20).
9: **end if**
10: Randomly sample $\eta \sim U[0,1]$.
11: **if** $\alpha \geq \eta$ **then**
12: Accept ν^* as $\nu_t^{(i+1)} = \nu^*$.
13: **if** ν^* is generated by *diffusion* **then**
14: Decrease diffusion variance δ_d^2.
15: **end if**
16: **else**
17: Reject ν^* and let $\nu_t^{(i+1)} = \nu_t^{(i)}$.
18: **end if**
19: **end for**
20: Return the new sample set $\{\nu_t^{(B+kM)} \mid k = 1, \ldots N\}$, and the estimated gait variable is $\hat{\nu}_t = (1/N)\sum_{k=1}^{N} \nu_t^{(B+kM)}$.
21: Estimate the current gait mode m_t with respect to $\hat{\nu}_t$ by using maximum-likelihood estimation as $m_t = \arg\max_{i=1,\ldots,R}\{N(\hat{\nu}_t; \mu_i, \delta_i^2)\}$.
22: Update the mean and variance for the current gait mode m_t by $\mu_{m_t} \leftarrow (1/2)(\mu_{m_t} + \hat{\nu}_t)$ and $\delta_{m_t} \leftarrow \delta_{m_t}^{1/(ct)}$, where c controls the annealing speed.
23: The diffusion variance is also updated by $\delta_d = \delta_{m_t}$.

VII. Experimental Results and Discussions

In this section, we will first discuss the experimental setup for training and testing. Second, we examine KGGM in terms of its capability of gait synthesis. Third, several particle-filtering-based inference algorithms are tested to show the advantage of SJD-MCMC. Fourth, our algorithm is compared with a set of state-of-the-art algorithms in detail. We also discuss the limitation of our algorithm.

A. Experimental Setups

1) Training Data Collection: The CMU Mocap library [4] provides a wealth of various human motion data, where we selected 20 walking motions (i.e., gait kinematics), represented by a series of Euler angles of joints, to learn KGGM. The gait appearances used for learning VGGM are generated by Autodesk MotionBuilder. We rendered 100 3-D gait animations by using the 20 gaits (that are used for KGGM training) and five human models (Fig. 7). Each 3-D gait animation was recorded under 12 camera views (30° apart).[3] Totally, we created 1200 30-frame gait animations (100 × 60). The whole training data set is available for public download (http://www.vcipl.okstate.edu/). Moreover, we extracted the binary silhouettes that were further "softened" by the signed distance transform used in [3] to create the gait appearances for VGGM training. Given a binary image containing one object, the signed distance transform assigns to each pixel, both inside (positive) and outside (negative) of the object, the minimum distance from that pixel to the nearest pixel on the border of the object. Such representation imposes smoothness of the distance between different gait appearances.

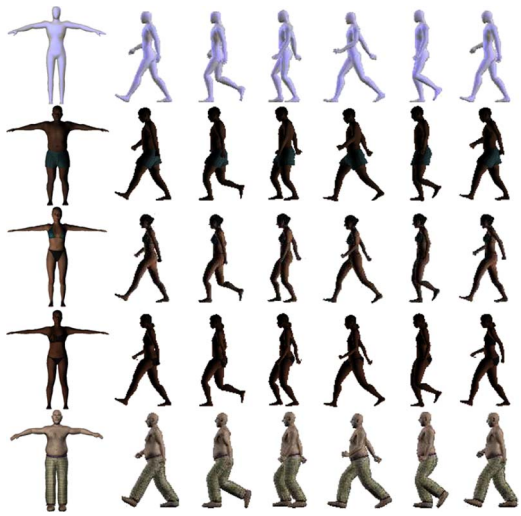


Fig. 7. Five 3-D shape models used in the experiment. The first and last ones are from the MotionBuilder Clip Art, and the others are from www.axyzdesign.com.

2) Testing Data Collection: We tested our algorithm on Subjects 1, 2, and 3 in the HumanEva-I data set [5]. We used the background subtraction technique in [61] to extract the foreground object.[4] We also developed two specific schemes to improve foreground extraction results. First, we divided each frame into two regions vertically according to the overall intensity value (roughly along the boundary of the carpet), and background subtraction is applied in each region independently. 2) We also employed some simple morphological operators to clean up the isolated foreground pixels and to fill the holes inside the body area. To extract the silhouettes, we need the 3-D/2-D hip positions in all frames (to be discussed shortly), the subject height, and the camera calibration information. Then, for each frame, the silhouette size is determined by the distance between the 3-D hip position and the camera as well as the subject height, and the silhouette center is the 2-D hip position computed from the 3-D hip position and the camera model. All extracted silhouettes that have different sizes have to be normalized to the size of training data (100 × 60). Some examples of normalized silhouettes of three subjects are shown in Fig. 8. Moreover, we need to apply the signed distance transform to convert binary silhouettes into grayscale gait appearances.

[3]We need a skeleton model to convert the motion data represented by Euler angles into joint positions (in millimeters) that are needed for animation generation. We chose one from the 20 CMU training gaits that best matches the five human models to generate all gait animations. All animations were created by setting the hip position as the image center.

[4]We used the C++ code from http://cvlab.epfl.ch/~tola/open_source.html.

Fig. 8. Silhouette sequences extracted for three HumanEva-I subjects.

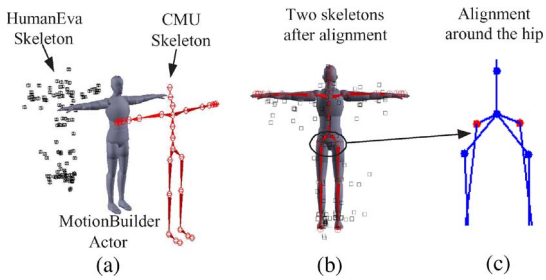


Fig. 9. Illustration of scaling/skeleton mappings. (a) Alignment of two skeletons after scale mapping. (b) Two skeletons after alignment. (c) The red and blue dots indicate different thigh configurations around the hip.

3) Local Error Analysis: Essentially, our algorithm computes the local motion that records the relative joint position with respect to the hip. The local error (ERR-I, in millimeters) measures the 3-D distance between the estimated and ground-truth joint positions that excludes the global position of the subject. Because the skeleton (or the marker system) in HumanEva-I (testing data) and the one in CMU (training data) have partially different joint configurations, we need to establish a mapping between them under the T-pose in order to compute a valid ERR-I value. Specifically, two operations are needed, namely, *scale mapping* and *skeleton mapping*, as shown in Fig. 9.

1) *Scale mapping* accounts for the height difference between the training and testing subjects, and it resizes the training skeleton to match the testing one according to their height ratio. This operation is based on the assumption that the lengths of all body parts are proportional to the height [37]. After scale mapping, the eight major joints (two elbows, hands, knees, and feet) are usually matched very well because they are commonly shared by most skeleton systems. However, the rest of the six joints (head, chest, two shoulders, and two thighs) are defined quite differently between the training and testing skeletons that need to be aligned further.
2) *Skeleton mapping* is inspired by the motion retargeting technique in computer graphics [62] that can adapt one motion data captured from one figure to another. This operation can be implemented efficiently via MotionBuilder. We first define a reference human model by which we can align two skeletons by sharing the same hip point. Then, for each of the six joints, we compute the translational displacement between two skeletons by which we can associate the same joint in two skeletons. This operation assumes that the relative position between each joint to the hip is fixed, which may not very accurate if some local deformation occurs. This step can be avoided if we can use the same skeleton for training and testing.

TABLE I
EVALUATION OF KGGM FOR GAIT SYNTHESIS

Joints (ERR-I: mm)	Single gait w/o mapping	Single gait with mapping	20 gaits with mapping
Head	209.96	121.15	67.35
Chest	40.34	26.39	12.04
R-shoulder	60.82	23.56	9.82
R-elbow	65.10	53.45	17.89
R-hand	200.46	103.84	46.63
L-shoulder	42.60	21.50	8.58
L-elbow	97.76	84.90	33.51
L-hand	187.63	94.65	33.82
R-thigh	77.89	13.78	9.24
R-knee	63.54	59.14	33.83
R-foot	117.67	93.80	66.33
L-thigh	80.59	15.83	8.86
L-knee	73.68	58.41	37.13
L-foot	120.84	115.32	65.81
Ave. Error	**102.77**	**63.26**	**32.20**

Once the kinematics of a new gait are estimated via VGGM and KGGM, they are initially represented by a sequence of Euler angles of joints that can be converted into relative joint positions with respect to the hip position by using the training skeleton. Then, via scale/skeleton mappings, the joint positions under the training skeleton can be mapped to the ones under the testing skeleton. ERR-I for each joint can be computed by aligning the estimated hip and the ground-truth one. ERR-I is mainly used to validate the usefulness of KGGM and VGGM.

4) Global Error Analysis: The global error (ERR-II) computes the distance between the estimated joint positions and the ground-truth ones in the global 3-D space. ERR-II is used for the overall performance evaluation of video-based human motion estimation. Since our algorithm directly outputs the local motion (centered with the hip) for each frame, we need to convert it into a global motion by the following three steps.

1) *Global hip localization* is implemented by a simple particle filter, given a series of gait silhouettes. The particle dynamics is a 3-D motion model that has a constant angular velocity and random walks in other two dimensions. The tracker is initialized by giving the ground-truth 3-D hip positions in the first frame with zero velocity. For a new frame, the hypotheses of 3-D hip position are generated by the motion model, which are mapped to the 2-D hypotheses in the image plane via the camera model. Then, we apply the previous silhouette to find the best 2-D/3-D hip hypothesis. We use only the upper part of the silhouette for matching, where the quality of foreground extraction is more stable. In our experiment, the average errors (in millimeters) of global hip localization are 21.75, 16.67, and 18.40 for Subjects 1, 2, and 3, respectively.
2) *Local motion estimation* is accomplished by using the proposed algorithm for a given gait appearance extracted according to the estimated 2-D/3-D hip positions, as discussed in Section VII-A-2. Local motion records the relative position between each joint and the hip.

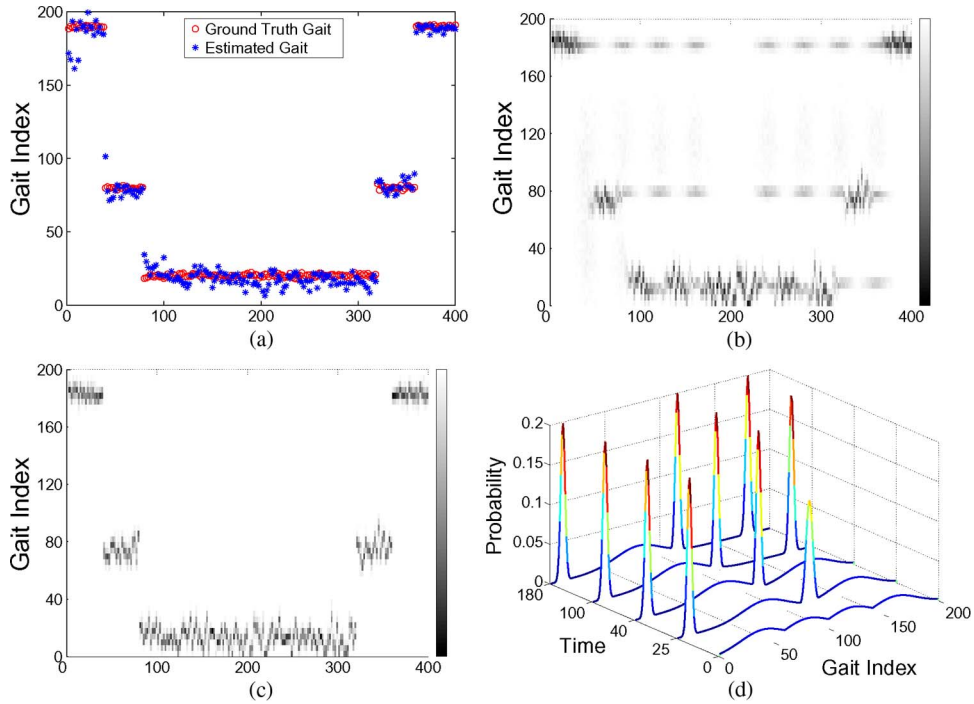


Fig. 10. Gait estimation results for Subject 1, where the horizontal/vertical axes show the frame index and the gait (interpolated from 20 training gaits along the gait manifold), respectively. (a) Comparison between the estimated and ground-truth gait values. (b) Distribution of the generated gait samples. (c) Distribution of the accepted samples. (d) Online learning results of the four gait modes defined in (16).

3) *Gait motion direction* is computed by comparing the present and previous 3-D hip positions. It is needed to impose the local motion estimation result (after scale/skeleton mappings) onto the 3-D hip position to reconstruct the complete 3-D motion estimation that records the global joint positions.

TABLE II
COMPARISON OF FOUR INFERENCE ALGORITHMS (ERR-I, IN MILLIMETERS)

ERR-I (mm)	LEB	Alg-1	Alg-2	Alg-3	Alg-4
Subject 1	32.20	95.54	89.84	83.46	**78.79**
Subject 2	46.25	100.26	92.86	85.01	**82.11**
Subject 3	44.87	103.32	94.31	91.72	**87.37**

B. Experiments on KGGM

We tested KGGM in terms of its ability to synthesize the unknown gait kinematics. According to (3), given a new gait coefficient along the kinematic gait manifold $\boldsymbol{\kappa}' \in \mathcal{M}_\kappa$ defined in (7), KGGM can synthesize the corresponding gait kinematics for an arbitrary pose by (3). We compared three methods of gait synthesis for Subject 1. The first is to directly use the best matched training gait (the smallest ERR-I) without scale/skeleton mappings. The second applies scale/skeleton mappings to correct the best matched training gait. The third is to use KGGM to approximate this unknown gait by an exhaustive search along $\boldsymbol{\kappa}' \in \mathcal{M}_\kappa$ and find the optimal gait coefficient that yields the best synthesized gait with the smallest ERR-I. The numerical results are shown in Table I, where we compare the three methods in terms of ERR-I of 14 joints (excluding the hip). It is clearly shown that KGGM provides much more accurate gait synthesis results. We call the smallest ERR-I provided by KGGM the *lower error bound* (LEB), which indicates the best performance that we can achieve.

C. Evaluation of the SJD-MCMC Inference Algorithm

1) Segmental Gait Modeling: Given the motion data of Subject 1, we first derived the optimal gait value (with the least ERR-I) for each frame using an exhaustive search along the gait/pose manifolds in KGGM that serves as the ground-truth value for algorithm evaluation. Then, we tested SJD-MCMC regarding its performance of dynamic gait estimation based on the observed gait appearances. As shown in Fig. 10(a), the segmental variability of the gait variable is evident, and the gait estimation results are quite accurate. Specifically, Fig. 10(b) and (c) shows the distribution of the gait samples generated before and after evaluation during SJD-MCMC. The mixed dynamics, in conjunction with segmental modeling, well capture the dynamic nature of the gait variable. Fig. 10(d) shows the learning of four modes in SJD-MCMC, among which only three modes are updated over time.

2) Local Motion Estimation: We implemented and evaluated four inference algorithms for gait estimation in Table II. Specifically, Alg-1 is the basic particle filter without segmental gait modeling. Alg-2 is the offline version of Alg-1 with the gait variable fixed to be the one estimated by Alg-1. In other words, Alg-1 and Alg-2 do not consider dynamic gait estimation. Alg-3 is the particle filter embedded with SJD-MCMC with online-mode learning, while Alg-4 is the offline version of Alg-3 and uses the four modes prelearned by Alg-3 for dynamic gait estimation. The ERR-I results improve from Alg-1 to Alg-4 progressively, showing that segmental gait modeling clearly improves the accuracy of motion estimation.

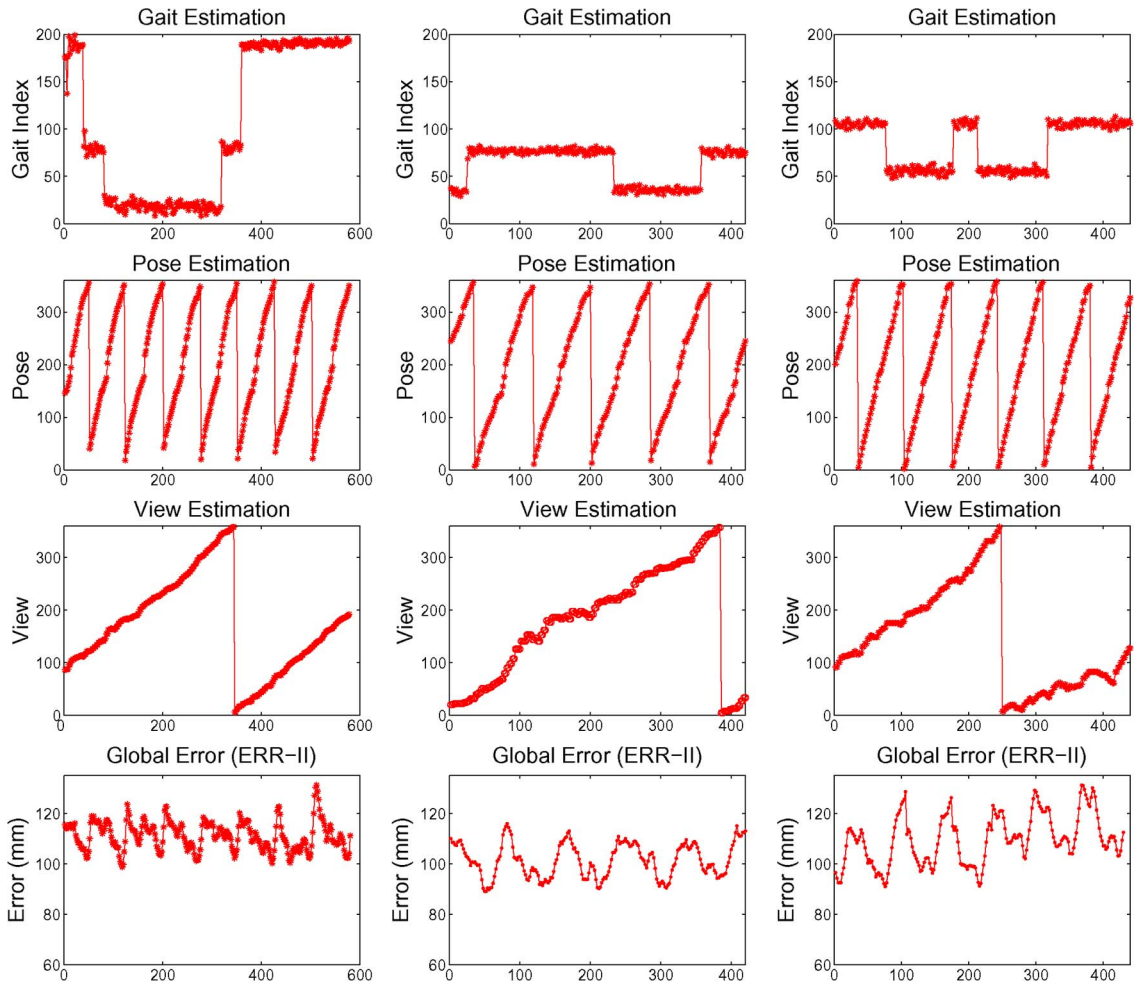


Fig. 11. Detailed results of (from left to right) Subjects 1, 2, and 3.

3) Detailed Result Analysis: Fig. 11 shows the estimated gait, pose, view, and ERR-II for three subjects, where we have the following four important observations: 1) The gait variable of three subjects exhibit obvious segmental variability; 2) the estimated pose and view well reflect the circular and periodic walking patterns; 3) three ERR-II plots roughly exhibit a periodic pattern that is consistent with the nature of a gait motion; and 4) the result Subject 3 is the worst one in terms of both ERR-I (Table II) and ERR-II (Fig. 11). It is mainly because Subject 3 is leaning inward during walking (more than other two subjects), while all training gaits follow a straight-line motion with an upright posture. We show more visual results in Fig. 12, where the ground-truth (in red) and estimated (in green) joint positions are plotted. In most frames, the motion estimation results are fairly accurate.

4) Sensitivity to the Training Data: We examined our algorithm (Alg-3) regarding its sensitivity to the number of training gaits and that of shape models, as shown in Table III (1S and 5S denote one and five shape models, respectively). It is obvious that the number of training gaits is critical to the accuracy of motion estimation, as indicated by both LEB and ERR-I. This is consistent with our assumption. It is also shown that the shape models are moderately important. Ideally, we could develop a *shape manifold* given sufficient training shapes that can provide rich shape modeling.

D. Overall Performance Evaluation

We compare our algorithm (Alg-4) with a series of recent algorithms in Table IV, where most methods were tested on HumanEva-I, some [34], [35] on HumanEva-II, and one [31] on both data sets (only the HumanEva-I result is reported here). HumanEva-II is more challenging due to the mixed motion types (walking and jogging) in a gait sequence, but the results from [34] and [35] listed here are only for the walking portion from Subject 2 who is also included in HumanEva-I. Furthermore, all algorithms are discussed in three groups according to how motion data are used for training. Group-I algorithms require motion data for training, and the same subject is used for training and testing. Group-II ones do not require any motion data for training. Group-III ones need motion data for training, and the subjects used for training do not include the ones used for testing. Our algorithm belongs to Group III. Additionally, we evaluate all algorithms in terms of experimental settings, including the number of cameras; global or local motion estimation, new testing subjects, learning approaches, and visual observations.

1) Group I: Essentially, the algorithms in this group aim at pose estimation instead of motion estimation due to the fact that the same subject is involved for training and testing. Most algorithms involve certain temporal prior in inference to

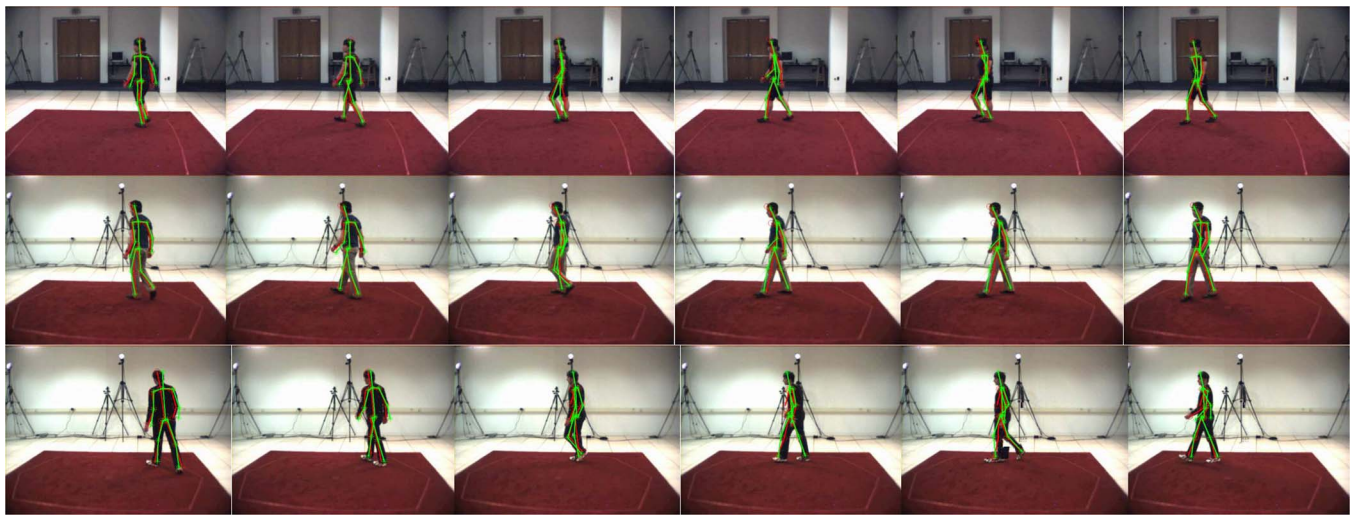


Fig. 12. Illustration of experimental results on three HumanEva-I subjects (from the first to the third row: Subjects 1, 2, and 3). The (as green "$*$") estimated and (as red "o") ground-truth joint positions are drawn on each image.

TABLE III
RESULTS OF DIFFERENT SETTINGS OF TRAINING DATA (ERR-I, IN MILLIMETERS)

	Single best-matched gait			10 Gaits			20 Gaits		
	LEB	1S	5S	LEB	1S	5S	LEB	1S	5S
Sub 1	63	132	123	45	111	106	32	90	83
Sub 2	80	146	130	66	123	116	46	98	85
Sub 3	79	150	137	67	122	109	45	105	92

TABLE IV
COMPARISON OF RECENT ALGORITHMS TESTED ON THE HUMANEVA-I OR HUMAN-II DATA SET ($*$)

	Sub. One	Sub. Two	Sub. Three	Camera Number	Global /Local	Motion Training	New Test Subjects	Discriminative /Generative	Visual Observations
Elgammal [17]	24.71	31.16	38.21	1	L	Y	N	G	silhouette
Poppe [14]	37.54	40.09	55.25	3	L	Y	N	D	HOG
Howe [15]	99			1	G	Y	N	D	silhouette+optical flow
Urtasun [11]	31.4	19.3	47.4	1	L	Y	N	D	HOG+edge
Sigal [12]	64.63			3	G	Y	N	D	shape context
Okada [63]	41.19	35.03	37.69	1	L	Y	N	D	HOG
Ni [64]		8.57		7	G	Y	N	G	silhouette
Bo [13]	23	13.7	40.3	1	L	Y	N	D	shape context
Gall [35]		32.23*		4	G	N	N	G	silhouette
Brubaker [23]	104			1	G	N	Y	G	edge
Mundermann [18]	53.1			7	G	N	Y	G	silhouette
Husz [31]	105.7			3	G	N	Y	G	silhouette
Canton-Ferrer [19]		115.21		4	G	N	Y	G	silhouette
Vondrak [42]		93.4		3	G	N	Y	G	silhouette
Xu [32]	140.35	149.37	156.3	4/7	G	Y	N	G	silhouette+edge
Cheng [34]		125*		4	G	Y	Y	G	silhouette
Peurum [30]	85.5	116.9	84.7	3	G	Y	Y	G	silhouette+edge
Our algorithm	111.19	105.43	113.28	1	G	Y	Y	G	silhouette
Our algorithm	87.58	99.04	105.73	1	G	Y	Y	G	silhouette (cleaned)

ensure smooth and continuous pose estimation. Specifically, discriminative approaches, e.g., [11]–[15] and [63], involve a direct mapping between visual observations to body configurations without explicitly pose modeling. The key issue of this mapping is how to handle the multiple-to-one problem due to the ambiguity of visual observations. On the other hand, generative approaches require explicit pose modeling for visual or kinematic data, where the key issue is how to deal with the view variability of visual observations. For example, the method in [17] involves a torus-shaped manifold for multiview pose modeling. Two mappings are learned to map both visual and kinematic data onto the torus. Then, the pose and view can be jointly estimated via maximum *a priori* estimation along the torus manifold. The best pose estimation result was reported in [64], where a hybrid sample-and-refine framework was proposed by combining both stochastic sampling and deterministic optimization for pose estimation and which also requires seven cameras.

2) Group II: Most algorithms in this group are generative approaches, since they usually need an explicit human model based on which motion estimation is accomplished. They can deal with unknown testing subjects, except the one in [35] that reports the best result in Group II and requires the 3-D body scan data of the testing subject. The human shape model plays a key role for Group-II algorithms, which is expected to be flexible and general enough to handle different subjects. For example, a physics-based biomechanical model was used in [23], [42]. In [18], a large set of body scan data was collected and used to learn a parametric 3-D human model that is general enough to handle various body shapes. Additionally, physical constraints can be incorporated into the human model, which provide useful priors for estimation and inference [19], [31]. The anthropometric measurements are also crucial, which can be learned online by using different shape models, such as cylinders [42] or visual hulls [18], [19], [31].

3) Group III: Similar to Group II, most algorithms in this group are generative approaches, which need explicit shape and motion modeling. For example, the body shape can be modeled by bounding boxes [30], visual hulls [34], cylinders [32], or 3-D character models (like ours). Motion modeling can be implemented by using a graphical model [30], [34] or a DR method (like ours) that can be trained from a set of kinematic data. In [32], the symmetric property of gait kinematics was used for motion modeling. Most algorithms model the motion and shape separately, and the two models are only used together during inference. However, one highlight of our paper is that the two models are integrated together via VGGM and KGGM during both learning and inference. We tested our algorithm on both actual observations and ones with manual cleanup, showing both the real performance and potential of our approach. A significant improvement is observed for Subject 1 (over 20 mm) compared with Subject 2 (6.4 mm) and Subject 3 (7.5 mm). It is because more improvement on foreground segmentation is obtained for Subject 1 after manual cleanup. Overall, our algorithm provides very promising results compared with the peers in the same group, considering that only a single camera is used.

E. Limitations

The following are some limitations of the proposed algorithm, which will guide our future research.

1) It is mainly designed for a specific type of motion, such as walking, and may not handle well different motion types (such as walking and jogging) together due to the fact that the strong dissimilarity between them may not be generalized well by one generative model.
2) The algorithm accuracy heavily relies on the quality and richness of the training gaits. Although increasing training gaits would improve the algorithm performance, it will also drastically increase the computational complexity.
3) There are a couple of systematic errors in this algorithm. One is that we ignore the local variability of a gait by assuming that any gait can be approximated locally by a straight motion with an upright posture. This assumption may not be accurate when the subject exhibits some non-straight or inclined motion patterns. The other is about the two assumptions made for the scale/skeleton mappings, which is needed to compute ERR-II for performance evaluation. This error could be reduced if the same marker system is used for training and testing.
4) Both VGGM and KGGM are used for whole-body modeling and are unable to provide detailed part-level modeling, limiting the accuracy of gait estimation and synthesis.
5) The computational load of the proposed algorithm is quite high due to the complexity of the generative models. Particularly, VGGM involves four parameters that have to be estimated during inference, where each Monte Carlo run involves a tensor product for the synthesis of gait appearance, and gait estimation is implemented as two-layer inference. Our algorithm was developed in Matlab 2009, and the current implementation (without program optimization) is about 35 s/frame (including both global hip localization and local motion estimation) on a PC computer (2-GB memory, dual-core, 2.2 GHz).

VIII. Conclusion and Future Research

We have presented a new approach to video-based human motion estimation that involves two gait generative models, namely, KGGM and VGGM, which represent gait kinematics and gait appearances, respectively, by a few latent variables. The two models have been synchronized by sharing the same pose manifold. The concept of gait manifold has been proposed to represent the variability among the training gaits by which two generative models can be integrated, so that we can infer the unknown kinematics of a new gait from its appearances via KGGM and VGGM. A new particle-filtering-based inference algorithm has been proposed for dynamic gait estimation along the gait manifold that is able to capture the segmental variability of the gait variable in a long sequence.

One straightforward way to enhance the algorithm performance is to improve the diversity and representativeness of the training gaits both visually and kinematically. However, the complexity of the dual generative models would increase exponentially as the training data grow. Therefore, our future research will focus on three issues: 1) how to create an adaptive yet simple human model to accommodate more shape variability; 2) how to span a more informative gait manifold that can improve the accuracy of gait synthesis under limited training data; and 3) how to learn a single gait manifold that can be shared by the dual gait generative models.

Acknowledgment

The authors would like to thank D. Biswas for generating the training data. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions that improved this paper.

REFERENCES

[1] L. Mudermann, S. Corazza, and T. P. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," *J. Neuroengineering Rehabil.*, vol. 3, no. 6, 2006.

[2] A. Elgammal and C.-S. Lee, "Separating style and content on a nonlinear manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 478–485.

[3] C.-S. Lee and A. Elgammal, "Modeling view and posture manifolds for tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[4] CMU Human Motion Capture Database. [Online]. Available: http://mocap.cs.cmu.edu

[5] L. Sigal and M. Black, "HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion," Brown Univ., Providence, RI, Tech. Rep. CS-06-08, 2006.

[6] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Underst.*, vol. 104, no. 2, pp. 90–126, Nov. 2006.

[7] R. Poppe, "Vision-based human motion analysis: an overview," *Comput. Vis. Image Underst.*, vol. 108, no. 1/2, pp. 4–18, Oct./Nov. 2007.

[8] A. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 882–888.

[9] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 44–58, Jan. 2006.

[10] K. Gauman, G. Shakhanarovich, and T. Darrell, "Inferring 3D structure with a statistical image-based shape model," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 641–647.

[11] R. Urtasun and T. Darrell, "Sparse probabilistic regression for activity-independent human pose inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[12] L. Sigal, R. Memisevic, and D. J. Fleet, "Shared kernel information embedding for discriminative inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2852–2859.

[13] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast algorithms for large scale conditional 3D prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[14] R. Poppe, "Evaluating example-based pose estimation: experiments on the HumanEva set," in *Proc. CVPR 2nd Workshop Eval. Articulated Human Motion Pose Estimation*, 2007.

[15] N. R. Howe, "Recognition-based motion capture and the HumanEva II test data," in *Proc. CVPR 2nd Workshop Eval. Articulated Human Motion Pose Estimation*, 2007.

[16] R. Rosales and S. Sclaroff, "Estimating 3D body pose using uncalibrated cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2001, pp. 821–827.

[17] A. Elgammal and C.-S. Lee, "Tracking people on torus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 520–538, Mar. 2009.

[18] L. Mudermann, S. Corazza, and T. P. Andriacchi, "Markerless human motion capture through visual hull and articulated ICP," in *Proc. NIPS Workshop Eval. Articulated Human Motion Pose Estimation*, 2006.

[19] C. Canton-Ferrer, J. Casas, and M. Pardas, "Exploiting structural hierarchy in articulated objects towards robust motion capture," in *Proc. Conf. Articulated Motion Deformable Objects*, 2008, pp. 82–91.

[20] F. Guo and G. Qian, "Monocular 3D tracking of articulated human motion in silhouette and pose manifolds," *EURASIP J. Image Video Process.*, vol. 2008, no. 3, pp. 1–18, 2008.

[21] T. Jaeggli, E. Koller-Meier, and L. V. Gool, "Multi-activity tracking in LLE body pose space," in *Proc. Int. Conf. Comput. Vis./2nd Workshop Human Motion*, 2007, pp. 42–57.

[22] T. Tangkuampien and D. Suter, "Real-time human pose inference using kernel principal component pre-image approximations," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 599–608.

[23] M. Brubaker, D. Fleet, and A. Hertzmann, "Physics-based human pose tracking," in *Proc. NIPS Workshop Eval. Articulated Human Motion Pose Estimation*, 2006.

[24] H. Ning, T. Tan, L. Wang, and W. Hu, "Kinematics-based tracking of human walking in monocular video sequences," *Image Vis. Comput.*, vol. 22, no. 5, pp. 429–441, May 2004.

[25] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard, "Tracking loose-limbed people," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 421–428.

[26] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr, "Randomized trees for human pose detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[27] L. Sigal, A. Balan, and M. J. Black, "Combined discriminative and generative articulated pose and non-rigid shape estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1337–1344.

[28] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a pose: Tracking people by finding stylized poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 271–278.

[29] L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2041–2048.

[30] P. Peurum, S. Venkatesh, and G. West, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," *Int. J. Comput. Vis.*, vol. 87, no. 1/2, Mar. 2010.

[31] Z. L. Husz, A. Wallace, and P. Green, "Evaluation of a hierarchical partitioned particle filter with action primitives," in *Proc. CVPR 2nd Workshop Eval. Articulated Human Motion Pose Estimation*, 2007.

[32] X. Xu and B. Li, "Learning motion correlation for tracking articulated human body with a Rao-Blackwellised particle filter," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[33] X. Lan and D. Huttenlocher, "A unified spatio-temporal articulated model for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 722–729.

[34] S. Y. Cheng and M. M. Trivedi, "Articulated human body pose inference from voxel data using a kinematically constrained Gaussian mixture model," in *Proc. CVPR 2nd Workshop Eval. Articulated Human Motion Pose Estimation*, 2007.

[35] J. Gall, B. Rosenhahn, T. Brox, and H. P. Seidel, "Optimization and filtering for human motion capture—A multi-layer framework," *Int. J. Comput. Vis.*, vol. 87, no. 1/2, pp. 75–92, Mar. 2010.

[36] B. Rosenhahn, C. Schmaltz, and T. Brox, "Markerless motion capture of man–machine interaction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[37] L. Mundermann, S. Corazza, and T. P. Andriacchi, "Accurately measuring human movement using articulated ICP with soft-joint constraints and a repository of articulated models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–6.

[38] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, "Detailed human shape and pose from images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[39] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, and J. Rodgers, "SCAPE: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, Jul. 2005.

[40] X. Lan and D. Huttenlocher, "Beyond trees: Common-factor models for 2D human pose recovery," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 470–477.

[41] M. W. Lee and I. Cohen, "Proposal maps driven MCMC for estimating human body pose in static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. II-344–II-341.

[42] M. Vondrak, L. Sigal, and O. C. Jenkins, "Physical simulation for probabilistic motion tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[43] M. Brubaker, D. Fleet, and A. Hertzmann, "Physics-based person tracking using simplified lower-body dynamics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[44] M. Brubaker and D. Fleet, "The kneed walker for human pose tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[45] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[46] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[47] N. Lawrence, "Gaussian process latent variable models for visualization of high dimensional data," in *Proc. Adv. Neural Inf. Process.*, 2003, pp. 329–336.

[48] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1441–1448.

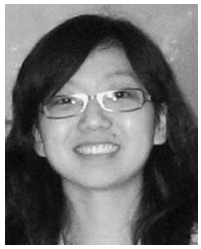
[49] R. Urtansun, "Motion model for robust 3D human body tracking," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2006.

[50] K. Moon and V. Pavlovic, "Impact of dynamics on subspace embedding and tracking of sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 198–205.

[51] N. Lawrence and J. Candela, "Local distance preservation in the GPLVM through back constraints," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 513–520.

[52] A. Gupta, T. Chen, F. Chen, D. Kimber, and L. Daivs, "Context and observation driven latent variable model for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[53] A. Elgammal and C.-S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 2, pp. 681–688.
[54] C.-S. Lee and A. Elgammal, "Body pose tracking from uncalibrated camera using supervised manifold learning," in *Proc. NIPS Workshop Eval. Articulated Human Motion Pose Estimation*, 2006.
[55] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua, "Priors for people tracking from small training sets," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 403–410.
[56] T.-P. Tian, R. Li, and S. Sclaroff, "Articulated pose estimation in a learned smooth space of feasible solutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, p. 50.
[57] C. H. Ek, P. Torr, and N. Lawrence, "Gaussian process latent variable models for human pose estimation," in *Proc. Mach. Learn. Multimodal Interaction*, 2007, pp. 132–143.
[58] C.-S. Lee and A. Elgammal, "Simultaneous inference of view and body pose using torus manifolds," in *Proc. Int. Conf. Pattern Recog.*, 2006, pp. 489–494.
[59] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2003, pp. 93–99.
[60] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 406–413.
[61] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 751–767.
[62] J.-S. Monzani, P. Baerlocher, R. Boulic, and D. Thalmann, "Using an intermediate skeleton and inverse kinematics for motion retargeting," *Comput. Graph. Forum*, vol. 19, no. 3, pp. 11–19, Sep. 2000.
[63] R. Okada and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 434–445.
[64] B. Ni, A. A. Kassim, and S. Winkler, "A hybrid framework for 3D human motion tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1075–1084, Aug. 2008.



Xin Zhang (S'06) received the B.S. degree in automatic control from Northwestern Polytechnical University, Xi'an, China, in 2003 and the M.S. degree in electrical engineering from Oklahoma State University, Stillwater, in 2005, where she is currently working toward the Ph.D. degree in the School of Electrical and Computer Engineering.

Her research interests include computer vision, machine learning, human motion analysis, and medical image processing.

Guoliang Fan (S'97–M'01–SM'05) received the B.S. degree in automation engineering from Xi'an University of Technology, Xi'an, China, in 1993, the M.S. degree in computer engineering from Xidian University, Xi'an, in 1996, and the Ph.D. degree in electrical engineering from the University of Delaware, Newark, in 2001.

From 1996 to 1998, he was a Graduate Assistant with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Since 2001, he has been with the School of Electrical and Computer Engineering, Oklahoma State University (OSU), Stillwater, where he is currently an Associate Professor. His research interests include image processing, computer vision, and machine learning.

Dr. Fan was awarded the First Prize in both the 1997 IEEE Hong Kong Section Postgraduate Student Paper Contest and the 1997 IEEE Region 10 (Asia–Pacific) Postgraduate Paper Contest. He was a recipient of the National Science Foundation CAREER Award and the Halliburton Excellent Young Teacher Award in 2004; the Halliburton Outstanding Young Faculty Award in 2006 from the College of Engineering, OSU; and the Outstanding Professor Award in 2008 from OSU–IEEE.