Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Modeling Style and Variation in Human Motion

Wanli Ma[1,2] Shihong Xia[1] Jessica K. Hodgins[3] Xiao Yang[1,2] Chunpeng Li[1] Zhaoqi Wang[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] Graduate University of Chinese Academy of Sciences
[3] School of Computer Science, Carnegie Mellon University

Abstract

Style and variation are two vital components of human motion: style differentiates between examples of the same behavior (slow walk vs. fast walk) while variation differentiates between examples of the same style (vigorous vs. lackadaisical arm swing). This paper presents a novel method to simultaneously model style and variation of motion data captured from different subjects performing the same behavior. An articulated skeleton is separated into several joint groups, and latent variation parameters are introduced to parameterize the variation of each partial motion. The relationships between user-defined style parameters and latent variation parameters are represented by a Bayesian network that is automatically learned from example motions. The geostatistical model, named universal Kriging, is extended to be a style-and-variation interpolation to generate partial motions for all joint groups. Experiments with sideways stepping, walking and running behaviors have demonstrated that the motion sequences synthesized by our method are smooth and natural, while their variations can be easily noticed even when their input style parameters are the same.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Animation

1. Introduction

Individually capturing the motions for all characters in a high density crowd animation is impractical. In most cases, a small number of motion templates are used, but motion clones [MLD*08] are easily noticed and distract from the quality of the animation. This problem can be overcome by generalizing motion data sets captured from different subjects, because people perform a behavior in a variety of different ways (variations) even if they intend to achieve the same goal (style). However, existing animation systems lack the ability to easily parameterize motions for different subjects. The main difficulty is the one-to-many mapping (see Figure 1) from a single set of user-defined style parameters to the variety of human motion seem with that single style. Generating realistic and appealing variations of a single style remains a challenging problem.

E-mails: mawanli@ict.ac.cn, xsh@ict.ac.cn, jkh@cs.cmu.edu, yangxiao01@ict.ac.cn, cpli@ict.ac.cn, zqwang@ict.ac.cn.

This paper presents a novel method to simultaneously model both style and variation in human motion. We normalize the example motions to use the same (standard) kinematic skeleton to unify their style parameter spaces. Then we separate this skeleton into four joint groups to allow greater generalization in the variations generated. For each joint group, a latent variation parameter is introduced to parameterize its variation. A Bayesian network (BN) is then constructed to describe the relationship between user-defined style parameters (such as stride length) and the latent variation parameters. We call this network a parameter propagation network. It can approximatively recover the dependencies between pairs of connective joint groups that have been lost in the skeleton separation. The universal Kriging model [HM71] is then enhanced to be a style-and-variation interpolation to generate partial motions for all joint groups.

The main contribution of our method is that it works well with motion data captured from different subjects and generates unlimited variants when given user-defined style parameters. The parameter propagation network ensures that our

DOI: 10.2312/SCA/SCA10/021-030

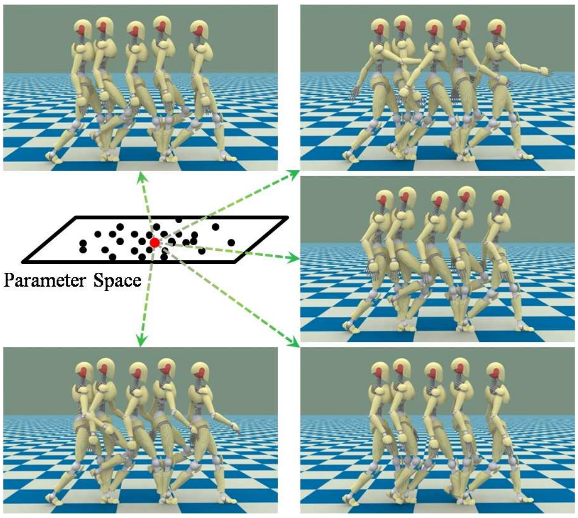


Figure 1: *The one-to-many mapping in walking. The red point indicates one user-defined style parameter (stride length), and there are five different example motions corresponding to this parameter. These motions vary in the details of their motion but the stride length is constant.*

model can create motions with variations even if their style parameters are identical. The latent variation parameters in this network can be automatically selected after users choose the style parameters. Therefore, our animation system can be used by novices to create realistic motions. Moreover, advanced users can manually specify the latent variation parameters for each joint group to achieve more detailed control of the motions. The computation for synthesizing short motion clips is fast and users can interactively control the parameters to generate new motions.

To evaluate our method, we have performed multiple experiments: (a) leave-one-out cross validation shows the accuracy of our method in predicting new motions; (b) an application for interactive motion synthesis is implemented to generate short motion clips in real time from user-defined control parameters; and (c) comparisons demonstrate that three long motion sequences synthesized by our method are visually different but more natural than the ones synthesized by several existing methods using the same example motion clips.

2. Related Work

Parameterization of motions is a powerful tool in predicting new motion styles from an existing motion database. Dimension reduction is one major branch of it, including principal component analysis (PCA) and Gaussian process latent variable models (GPLVMs). A coherent locomotion engine was established by using multilevel PCA [GBT04]. This engine is capable of extrapolation of physical parameters of locomotion. Urtasun and his colleagues made use of PCA on entire motions rather than poses so that they could approximate example motions and extrapolate realistic animations at different speeds or distances [UGB*04].

GPLVM is a probabilistic non-linear mapping from the embedded space to the data space, which was first introduced by Lawrence for visualization of high-dimensional data [Law03]. A SGPLVM was then adapted from this model for motion editing while maintaining its original style [GMHP04]. To express conditional independencies in motion data, Lawrence and Moore extended the GPLVM through hierarchies [LM07]. Wang et al. augmented the GPLVM to be a Gaussian process dynamical model (GPDM) with a latent dynamical model that enables predictions to be made about future data and helps regularize the latent space for modeling temporal data in general [WFH07].

Another major branch of motion generalization is motion interpolation, such as radial basis functions (RBFs) and the universal Kriging model. Motion interpolation makes it possible that users can synthesize new motions with custom control parameters. Kovar and Gleicher created a denser sampling of parameter space and applied blending techniques to generate new motions [KG04]. RBFs were used to produce motions "verbs" with the parameters "adverbs" and a "verb graph" was constructed to create smooth transitions between these actions [RBC98]. The universal Kriging model was first introduced to human animation by Mukai and Kuriyama, and was demonstrated to predict motions more accurately than RBFs do [MK05]. This model is most closely related to our work. However, these interpolation methods would not work for our problem because they could not produce variants given a single set of style parameters.

Variation is important to realistic crowd animations. The simplest way of generating variation is to add noise. The Perlin noise function is a type of gradient noise that is often used by visual effects artists to increase the appearance of realism in computer graphics. It can be used to create animations of running, standing and dancing using a noise function to move the limbs [Per95]. Bodenheimer and his colleagues constructed another noise function based on biomechanical considerations to introduce natural-looking variability into cyclic animations [BSH99]. However, these approaches require manual tuning of the parameters and do not guarantee that the generated motion will appear natural. In fact, biomechanical research have argued that variation is not just noise or error, but is a functional component of motion [HW98].

Many statistical methods have been proposed to model variation. Pullen and Bregler [PB00, PB02] approximated the correlations between the degrees of freedom (DOFs) in motion data with a distribution, and synthesized new motions by sampling from this distribution. Recently, dynamic Bayesian networks (DBNs) were introduced to model spatial and temporal variations in motion data [LBJK09]. Two DBNs were automatically constructed to capture the properties of conditional independence in "similar but slightly different" example motions. A transition network was learned to model subsequent frames given the previous two frames. The transition network would be repeatedly "unrolled" to synthesize new variants, but it would produce unnatural frames when generating motion sequences that are much longer than the example motion clips.

Assembling partial motions can greatly enrich the variations in motion database. Bruderlin and Calvert partitioned an articulated skeleton into lower limbs and upper body to synthesize motions by kinematic function [BC89]. Ikemoto and Forsyth introduced a technique for replacing the motion of some limbs with another motion, and suggested rules for synthesizing natural-looking motions [IF04]. Similarly, Jang et al. enlarged the motion database by analogous combination of partial motions [JLLL08]. They attempted to separate skeletons into more than two parts and created clusters of partial motions from which combinations can be selected. The main weakness of these methods is that the dependencies between different joint groups have been lost, so we construct a parameter propagation network to approximately recover these dependencies in our model.

3. Method

In this paper we present a method to simultaneously model style and variation of motion data captured from different subjects performing the same behavior. In fact, style and variation are two vital components of human animation. These two concepts have frequently been used but without a consistent definition. We define behavior as the kind of human action. For example, we consider walking and running as two different behaviors. Then we define style to be a continuous parameter space that intuitively determines the basic motion of a certain behavior. For example, stride length, velocity, length of double support, are all possible style parameters of walking. We define variations as the differences between motions of the same style. For example, some people may swing their arms further or pick up their foot higher during swing (Figure 1). These details do not change the fundamental style or pacing of the motion, but do change its appearance. In short, style differentiates between examples of the same behavior while variation differentiates between examples of the same style.

Figure 2 illustrates the work flow of our method. There are three phases: data preprocessing (§4), building a hierarchical model (§5) and motion synthesis (§6). In the preprocessing phase, all example motions are normalized to use the same (standard) kinematic skeleton. Space warping and time warping (§4.2) are then implemented to establish a correspondence for these example motions. After the users specify the control parameters $\{\mathbf{c}_i\}$, a hierarchy can be created in the modeling phase. We separate the skeleton into joint groups (§5.1) and introduce latent variation parameters $\{\xi_i\}$ to all joint groups (§5.2). Then we construct a parameter propagation network to describe the relationship between $\{\mathbf{c}_i\}$ and $\{\xi_i\}$ (§5.3), and build a partial style-and-variation interpolation (SVI) model for each joint group (§6.2). Given a new parameter $\mathbf{c}$, the hierarchical model can predict partial motions in the synthesis phase. By assembling these partial motions, a whole-body motion can be synthesized. Motion transitions are created if needed. The final motion can be obtained after eliminating footskate (§6.3).

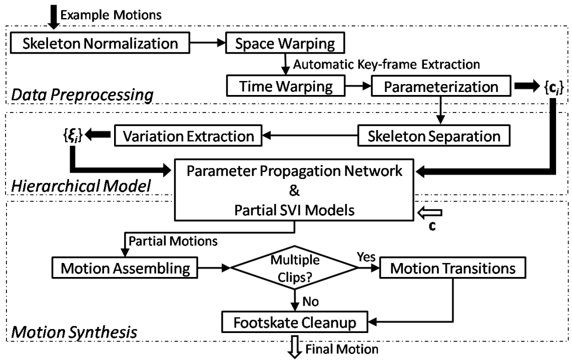


Figure 2: *The diagram of our method.*

4. Data Preprocessing

A human motion $\mathcal{M}$ consists of a sequence of poses: $\mathcal{M} = \{\mathbf{p}_1, ..., \mathbf{p}_T\}$, where T is the duration of the motion. Each pose $\mathbf{p}_i$ contains the global 3D position and orientation of the root node and the rotations of all the other joint nodes relative to their parent joint nodes. The global position $pos_j(t)$ and rotation $rot_j(t)$ of the j-th joint node at frame t can be easily computed with forward kinematics [JW02]. In our experiments, all rotations are represented by unit quaternions. We employ the sinusoidal rotational distance [PSS02] to measure the distance between two unit quaternions q_1 and q_2:

$$\|q_1 - q_2\|_s = sin(\|log(q_1^{-1} q_2)\|)$$

4.1. Automatic Key-Frame Extraction

To interpolate captured motions, we need to segment them into short clips and establish a correspondence for these clips in the temporal space, which is a time-consuming process. Automatic key-frame extraction would allow us to easily segment the motion sequence based on the key-frames. These key-frames can also be used to time-warp the example motion clips. Our criterion to extract key-frames is the space distribution of the joint nodes. The local minima and maxima of the volume of the bounding box that covers all joint nodes form the key-frames.

For different motion behaviors, e.g. boxing and walking, the importance of each joint node is not equal. Therefore, we consider the kinematic skeleton as two parts to extract key-frames: lower body (two legs) and upper body (the rest of the body). Given a motion sequence $\mathcal{M}$, we construct a global position matrix $\mathbf{P}(t)$ at every frame t and divide it into two submatrices $\mathbf{P}_l(t)$ and $\mathbf{P}_u(t)$ for each part:

$$\mathbf{P}(t) = \begin{bmatrix} \mathbf{P}_l(t) \\ \mathbf{P}_u(t) \end{bmatrix} = \begin{bmatrix} pos_1(t) \\ pos_2(t) \\ \vdots \\ pos_J(t) \end{bmatrix}_{J\times 3} \quad (1)$$

Then we combine the submatrices of each part at all frames into two matrices $\mathbf{P}_l$ and $\mathbf{P}_u$, and execute PCA to get two representative matrices:

$$\mathbf{P}_l^* = \begin{bmatrix} \mathbf{P}_l^*(1) \\ \vdots \\ \mathbf{P}_l^*(T) \end{bmatrix} \qquad \mathbf{P}_u^* = \begin{bmatrix} \mathbf{P}_u^*(1) \\ \vdots \\ \mathbf{P}_u^*(T) \end{bmatrix} \quad (2)$$

The quadratic sum of their eigenvalues can demonstrate the space distribution of each partial motion. Therefore, we define the traces of their covariance matrices as the importance of each part ($Tr(\cdot)$ is the trace of a matrix):

$$D_l = \sqrt{Tr(\mathbf{P}_l^* \mathbf{P}_l^{*\prime})} \qquad D_u = \sqrt{Tr(\mathbf{P}_u^* \mathbf{P}_u^{*\prime})} \quad (3)$$

Finally, a novel measurement is proposed to extract key-frames from motion sequences:

$$S_{\mathcal{M}}(t) = \begin{cases} \sqrt{Tr(\mathbf{P}_l^*(t)\mathbf{P}_l^*(t)')} & D_l \geq D_u \\ \sqrt{Tr(\mathbf{P}_u^*(t)\mathbf{P}_u^*(t)')} & D_l < D_u \end{cases} \quad (4)$$

$S_{\mathcal{M}}(t)$ is defined as a piecewise function because we only need the local minima and maxima inside a single motion. If D_l is larger than D_u, the space distribution of the lower body would be considered in selecting key-frames. Otherwise, we use the space distribution of the upper body to extract key-frames. Once $S_{\mathcal{M}}(t)$ is calculated from a motion sequence, key-frames can be rapidly located by detecting the local minima and maxima of $S_{\mathcal{M}}(t)$ with the technique proposed by Ik Soo and Thalmann [IST01] (see Figure 3).

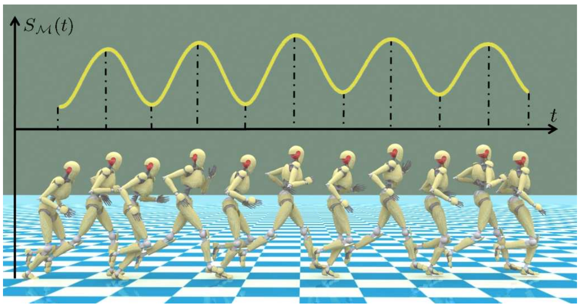


Figure 3: *Automatically extracted key-frames of running. Each local minima or maxima of curve $S_{\mathcal{M}}(t)$ corresponds to a key-frame.*

4.2. Motion Correspondence

With the extracted key-frames, we can divide each motion into several clips and create the proper correspondences based on the spatial and temporal components. The spatial components are influenced by the position and orientation of the root node, and the temporal components are determined by the speed of the movement.

Space Warping: To unify the movement direction of all example motion clips, we rotate them about the vertical axis (y-axis) such that the overall movement direction across the motion is as closely aligned (with the x-axis) as possible. For each motion clip $\mathcal{M}$, we denote its direction as $dir(\mathcal{M})$. Let $R(\theta)$ be the transformation matrix that rotates about the vertical axis by θ degrees. Then the space warping can be described as:

$$\theta^* = \arg\min_{\theta} \|R(\theta)dir(\mathcal{M}) - x\|^2 \quad (5)$$

$$\mathcal{M}^* = R(\theta^*)\mathcal{M} \quad (6)$$

Time Warping: Our database is composed of motions performed by many different subjects. As a result, the timings of these example motions vary. An Incremental Time Warping (ITW) technique [PSS02] is used to establish a correspondence and scale them to be of the same duration. This time warping technique ensures that the progression through the synthesized motion clips is monotonically increasing.

5. Hierarchical Model

We separate a kinematic skeleton into four joint groups. For each joint group, a latent variation parameter is introduced to parameterize the variation. Then a Bayesian network is constructed to describe the relationship between the user-defined style parameters and these latent variation parameters. The network can approximatively recover the dependencies between pairs of connected joint groups.

5.1. Kinematic Joint Groups

Modeling partial motions not only reduce the complexity of the models but also enrich the variations of the motion database. The more joint groups that are used, the more combinations become available, but it becomes more difficult to generate natural-looking motions. We partition the kinematic skeleton into four joint groups as shown in Figure 4: legs, left arm, right arm and the rest upper body. This set of partition has given good results in practice [JLLL08].

5.2. Latent Variation Parameters

To describe the variations between motions of the same style, we introduce a novel latent variation parameter. For

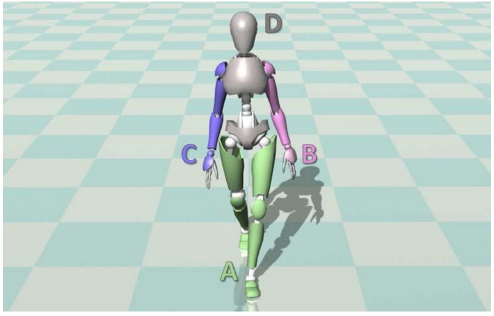


Figure 4: *Kinematic joint groups in an articulated skeleton. Each letter represents a joint group:* ***A*** *= legs,* ***B*** *= left arm,* ***C*** *= right arm, and* ***D*** *= upper body.*

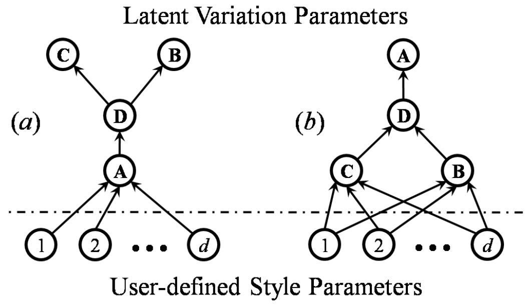


Figure 5: *Structures of parameter propagation networks: (a) lower-first structure; (b) upper-first structure.*

any motion $\mathcal{M}$, the variation of the j-th joint node can be defined as the sum of the sinusoidal rotational distances between every pair of sequential poses:

$$\delta_j(\mathcal{M}) = \sum_{t=1}^{T-1} \|rot_j(t+1) - rot_j(t)\|_s \tag{7}$$

The variance of $\delta_j(\mathcal{M})$ among all motions of the same behavior $\mathcal{C}$ represents the importance of the j-th joint node:

$$\eta_j = Var[\delta_j(\mathcal{M})]|_{\mathcal{M}\in\mathcal{C}} \tag{8}$$

For each joint group $I \in \{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$, we choose the variation of the most determinative joint node to parameterize the partial motion $\mathcal{M}^I$. Formally,

$$\tau(I) = \arg\max_{j\in I} \eta_j \tag{9}$$

Then we can define $\xi_I = \delta_{\tau(I)}(\mathcal{M})$ as the latent variation parameter that intuitively presents the movement range of the partial motion $\mathcal{M}^I$.

5.3. Parameter Propagation Network

The dependencies between the partial motions of the joint groups are lost when the motions are divided into partial motions. A Bayesian network, which we call a parameter propagation network, can be constructed to approximately recover these dependencies. The parameter propagation network represents the relationship between parameters instead of pose data as was used by Lau and his colleagues [LBJK09]. This network ensures that our method can deal with the one-to-many situations in motion database.

Denote the parameter propagation network as $G = (V,E,W)$. The node set $V = \{v_i\}$ is the union of the elements of d-dimensional user-defined style parameter $\mathbf{c} = \{c_1,...,c_d\}$ and all partial latent variation parameters $\{\xi_I\}$. Each edge (i,j) in the edge set E declares that $v_i \in \mathbf{Pa}(v_j)$, where $\mathbf{Pa}(\cdot)$ is the parent set of a node. The weight of edge (i,j) is defined as $w_{ij} \in W$, which represents the influence of node v_i on node v_j. Figure 5 illustrates two kinds of parameter propagation networks. For the motion behaviors that satisfy $D_l \geq D_u$ as mentioned in §4.1, user-defined style parameters are always highly related with the movement of the two legs, so we employ the lower-first structure; otherwise, we choose the upper-first one.

Given a new style parameter, we can generate the latent variation parameters of all joint groups according to the parameter propagation network:

$$v_j = \sum_{v_i\in\mathbf{Pa}(v_j)} w_{ij}v_i + \varepsilon_j, \quad \varepsilon_j \sim N(\mu_j,\sigma_j^2) \tag{10}$$

where $N(\mu_j,\sigma_j^2)$ is a normal distribution with mean μ_j and variance σ_j^2. The unknown variables $\{w_{ij}\}$, μ_j and σ_j^2 can be uniquely determined from the database by using a least squares approximation with a pseudo-inverse matrix. With this network, our model can create unlimited variations of a motion even when the style parameters remain constant.

6. Motion Synthesis

We enhance the universal Kriging model to be a style-and-variation interpolation to generate partial motions for all joint groups. Whole-body motions can be created by assembling these partial motions. When synthesizing long motion sequences, we need to create transitions between every pair of sequential clips. Finally, footskate cleanup is necessary to recover the global positions of the root node that have been discarded in motion interpolation.

6.1. Universal Kriging Model

The Kriging model, named for pioneer D.G.Krige, is a best linear unbiased prediction of a random function. Ordinary kriging requires that a condition, called *intrinsic stationary*, is satisfied [Cre93].

Intrinsic Stationary: A random function $S(\cdot)$ is intrinsic stationary, if any arbitrary pair of parameters $(\mathbf{c}_i, \mathbf{c}_j)$ satisfies

$$E[S(\mathbf{c}_i) - S(\mathbf{c}_j)] = 0, Var[S(\mathbf{c}_i) - S(\mathbf{c}_j)] = \gamma(\|\mathbf{c}_i - \mathbf{c}_j\|)$$

where $\gamma(\cdot)$ is a *variogram* function that describes the relationship between parameter distance and the variance of the distance of $S(\cdot)$.

This condition is hard to guarantee in many applications. Huijbregts and Matheron extend this model to be a universal one by assuming that a component of $S(\cdot)$ is unrelated to the random function [HM71]. In other words, $S(\cdot)$ is separated into a trend component $m(\cdot)$ and a residual component $r(\cdot)$. $m(\cdot)$ can be directly computed from the parameters and $r(\cdot)$ is a random function which satisfies the intrinsic stationary condition.

Denote $\mathbf{c}_i$ as the control parameter of the i-th sample s_i. Let $m_i = m(\mathbf{c}_i)$ and $r_i = s_i - m_i$ be the trend component and residual component of s_i respectively. Given a new parameter $\mathbf{c}$, we can predict the corresponding function value $S(\mathbf{c})$ as follows:

$$S(\mathbf{c}) = m(\mathbf{c}) + \sum_i \lambda_i(\mathbf{c}) r_i, \quad \sum_i \lambda_i(\mathbf{c}) = 1$$

where $\{\lambda_i(\cdot)\}$ are the weight functions, which can be estimated with the help of variogram function (please refer to [MK05] for the details):

$$\begin{bmatrix} \Lambda(\mathbf{c})' \\ \kappa \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{1}' \\ \mathbf{1} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma(\mathbf{c})' \\ 1 \end{bmatrix}, \mathbf{R} = \{\gamma(\|\mathbf{c}_i - \mathbf{c}_j\|)\}_{ij}$$

$$\gamma(\mathbf{c}) = [\gamma(\|\mathbf{c}_1 - \mathbf{c}\|), \gamma(\|\mathbf{c}_2 - \mathbf{c}\|), \dots, \gamma(\|\mathbf{c}_N - \mathbf{c}\|)]$$

where $\Lambda(\mathbf{c})$ is a row vector composed of weight functions $\{\lambda_i(\mathbf{c})\}$, $\mathbf{1} = [1, \dots, 1]$, κ is a Lagrange multiplier and N is the total number of example motion clips. Note that the inverse matrix in the linear equations can be calculated as a preprocessing step, so the weight functions $\{\lambda_i(\mathbf{c})\}$ can be estimated in real time.

6.2. Style-and-variation Interpolation

Mukai and Kuriyama proposed two motion interpolation models: per-element interpolation and per-pose interpolation [MK05]. In contrast to these models, we treat motion clips as the basic units instead of DOFs or poses. Therefore, our model can generate motion clips with only a few input parameters.

As mentioned in §5, we divide the kinematic skeleton into four joint groups. Given a user-defined style parameter $\mathbf{c}$, we can automatically generate the latent variation parameters $\{\xi_I\}$ for all joint groups with the parameter propagation network. Then the partial motions $\mathcal{M}^I$ $(I \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\})$ can be synthesized as follows:

$$\begin{aligned} \mathcal{M}^I(\mathbf{c}) &= m^I([\mathbf{c}, \xi_I]) + \sum_i \lambda_i^I([\mathbf{c}, \xi_I]) r_i^I \\ &= m_s^I(\mathbf{c}) + m_v^I(\xi_I) + \sum_i \lambda_i^I([\mathbf{c}, \xi_I]) r_i^I \end{aligned} \quad (11)$$

where $m_s^I(\cdot)$ and $m_v^I(\cdot)$ are the trend components of style and variation respectively. In our experiments, these two components are defined as two hyperplanes:

$$m_s^I(\mathbf{c}) = \alpha_0^I + \sum_{i=1}^{d} \alpha_i^I c_i \quad (12)$$

$$m_v^I(\xi_I) = \beta_0^I + \beta_1^I \xi_I \quad (13)$$

The coefficients $\{\alpha_i^I\}$ and $\{\beta_i^I\}$ can also be determined by using the least squares technique. In addition, the algorithms of estimating variogram function and calculating weight functions are directly inherited from the universal Kriging model.

6.3. Post Processing

We perform two forms of post processing: transition creation and footskate cleanup to compute the motion of the root node.

Motion Transitions: Let $\mathcal{M}_1 = \{\mathbf{p}_1^1, \dots, \mathbf{p}_T^1\}$ and $\mathcal{M}_2 = \{\mathbf{p}_1^2, \dots, \mathbf{p}_T^2\}$ be two synthesized whole-body motion clips (both have T frames). To stitch them together, a displacement mapping technique is employed [BW95]. We only edit $\mathcal{M}_2$ to make a smooth transition while maintaining the details of $\mathcal{M}_1$. Suppose the transition lands on the t-th frame of $\mathcal{M}_2$. Then a new motion can be created as:

$$\begin{aligned} \widetilde{\mathcal{M}} &= \mathcal{M}_1 \oplus \mathcal{M}_2 \\ &= \{\mathbf{p}_1^1, \dots, \mathbf{p}_{T-1}^1, \mathbf{q}_1, \dots, \mathbf{q}_t, \mathbf{p}_{t+1}^2, \dots, \mathbf{p}_T^2\} \end{aligned}$$

where

$$\mathbf{q}_i = \mathbf{p}_i^2 + \rho_i \Delta\mathbf{p}, \quad \Delta\mathbf{p} = \mathbf{p}_T^1 - \mathbf{p}_1^2 \quad (14)$$

$$\rho_i = 2\{\frac{i-1}{t-1}\}^3 - 3\{\frac{i-1}{t-1}\}^2 + 1 \quad (15)$$

The blend weight ρ_i is inspired by the one used in motion graphs [KGP02]. It simultaneously satisfies $\rho_1 = 1$ and $\rho_t = 0$. The landing frame t in $\mathcal{M}_2$ is usually chosen as the one that reaches the first local maximum of the pose distance to the last frame of $\mathcal{M}_1$.

Footskate Cleanup: The root positions of the synthesized motion have been set to the origin in the synthesis process. The global positions of the joint nodes for the two feet can be computed with forward kinematics. For behaviors without flight phases, such as walking, the foot that contacts the ground can be detected by comparing the height of the two feet. Then the root position can be reconstructed by fixing this foot on the ground and treating it as the root of the kinematic skeleton [TLP07]. For other motion behaviors (e.g. running), however, the vertical position of the root node needs to be included in the partial model **A** (legs). When generating a new motion, the corresponding vertical position

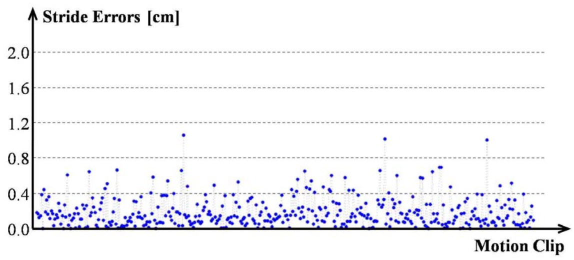


Figure 6: *Errors between the stride length of the validation walking clips and the predicted motion clips via leave-one-out cross validation.*

curve is simultaneously predicted. If at least one foot stays on the ground, the root positions change in the same way as the behaviors without flight phases. Otherwise, if both feet are higher than the ground or a specified threshold, the predicted curve will be employed to decide the vertical root positions and the horizontal root positions will change at a constant speed.

7. Results

In our experiments, example motion sequences are captured at 120 frames per second from 25 different subjects. The kinematic skeletons of these subjects are composed of 32 joint nodes, including 6 virtual joint nodes. There are three behaviors in our motion database: sideways stepping, walking and running.

7.1. Model Evaluation

Leave-one-out cross validation is employed to evaluate the accuracy of our model. As the name suggests, it involves using a single example motion clip from the database as the test data and the remaining clips as the training data. The process is repeated until each example motion clip in the database has been used once as the test data. Figure 6 shows the result for walking. The maximum error in stride length is below 1.2 cm. Therefore, the test motion clips have been accurately predicted according to their style parameters.

7.2. Interactive Motion Synthesis

Due to the calculation of the high-dimensional inverse matrix, the process of training each partial model is time-consuming. However, new motion clips can be quickly synthesized by solving the linear equations as described in §6. An animation system has been implemented for interactive motion synthesis with our model (Figure 7(a)). When users drag the red point to a new position, the motions for all kinematic skeletons in the current scene are updated in real time. The axes of the control panel represent the first and

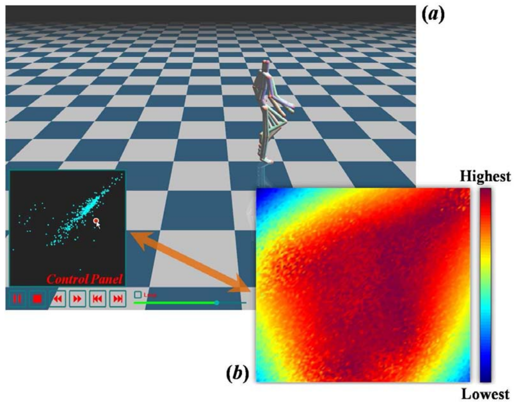
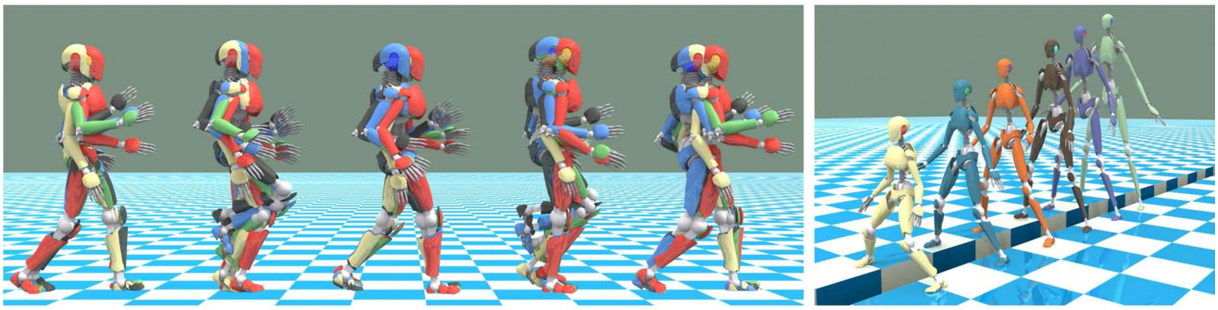


Figure 7: *(a) User interface of the interactive motion synthesis application. The points in the control panel indicate the user-defined style parameters of example motion clips. (b) Reliability map of the style parameters. Red and blue areas indicate regions of high and low reliability respectively.*

the second dimension of the user-defined style parameter respectively. For example, the stride lengths of the left support phase and the right support phase are the two elements of a style parameter for the walking behavior. Figure 7(b) shows a map of reliability in the style parameter space, indicated by colors. It demonstrates that the distances between the user-specified style parameter $\mathbf{c}$ and the example style parameters $\{\mathbf{c}_i\}$ have a strong correlation with the reliability level. Moreover, it demonstrates that our model can even extrapolate plausible motions in some parameter regions that are not covered by the example motion clips.

7.3. Variation Generation

Our model can generate variants even with the same user-defined style parameter, which is impossible for the existing interpolation algorithms. Figure 8(a) presents five synthesized walking motion clips for a single skeleton with the same stride length. The differences between these motion clips are very easy to notice, while their stride lengths coincide with the user-specified style parameter. Our model can also be adapted to create realistic motions for different skeletons by normalizing the input style parameters by their kinematic parameters. Figure 8(b) illustrates a synthesized sideways stepping across a crevasse for six different sizes of skeletons. To fix the problem of motion clones in crowd animation, we create 16 different subjects with random stature, and synthesize a long walking sequence for each subject using the same style parameter (speed). The variations between these synthesized motions are visually apparent. In the example, all subjects are arranged in a circle indicating that their speed constraints have been satisfied (please refer to the accompanying video).



(a) **(b)**

Figure 8: *Variations in synthesized motions (shown in different colors) of the same style that is controlled by user-defined parameters: (a) key-frames of five walking clips for a single skeleton given the same stride length, and (b) sideways stepping for six different sized characters across a crevasse.*

7.4. Comparisons with Related Methods

We use the five example motion clips shown in Figure 1 to compare our model with several methods for generating natural long motion sequences (much longer than the example motion clips). These example motion clips vary but have approximately the same style parameters (stride length). Adding Perlin noise [Per95] to the rotations of an existing motion sequence (generated by randomly combining these walking clips) is one of the simplest methods. However, the resulting motion sequences seem unnatural even with manually tuned parameters. GPLVM [Law03] can be used to describe the nonlinear mapping from the latent variable space to the pose data space. We learn a GPLVM from the poses of these five example motion clips, but its low-dimensional latent variable space is not intuitive for the creation of motion sequences. Therefore, we roughly synthesize a motion sequence with a helix trajectory in the latent variable space. As expected, jerks appear in this motion sequence because of unreasonable latent variables.

RBFs [RBC98] and the universal Kriging model [MK05] cannot produce variants with the same style parameters. The radial functions in RBFs return constant values with the same parameters. Similarly, the trend component and the weight functions of universal Kriging model also return constant values. Lau et al. proposed two DBNs to model the variation in "similar but slightly different" motion data [LBJK09]. After learning two DBNs from the five example motion clips, the so-called transition network is repeatedly "unrolled" to create long motion sequences. However, unnatural frames often appear after many iterations when generating motion sequences that are much longer than the example motion clips. In contrast to these related methods, our model creates three natural long motion sequences that are visually different but with consistent style parameters. Please refer to the accompanying video for the resulting animations.

8. Discussion

We present a novel method to model style and variation in motions of the same behavior. In our model, an articulated skeleton is divided into four joint groups. Partial style-and-variation interpolation models are built for each joint group, and the dependencies between them are described as a parameter propagation network. This network ensures that we can create motions with variations even if the style parameters are constant.

In the skeleton separation process, we put the two legs in one joint group but divide the arms into two joint groups. This is because foot constraints are very important to motion synthesis. If we generate the partial motion for each leg separately, there is no guarantee that the foot constraints can be maintained. As a result, artifacts will be easily noticed. However, the two arms move independently, so we can treat them as two different joint groups to enrich the variation.

Our model can work well with example motions from different subjects. To overcome the one-to-many problem (Figure 1), latent parameters are introduced to describe the variations. They can be automatically generated according to the parameter propagation network after the user specify the style parameters. Therefore, our animation system allows a novice to create realistic motions. Moreover, advanced users are also allowed to manually assign the latent variation parameters for each joint group to synthesize motions as they wish.

The runtime for synthesizing short motion clips is very efficient. Users can interactively control the style parameters to generate new motions. To synthesize long motion sequences, however, motion transitions must be employed. The computational cost at runtime is proportional to the duration of the sequence. On average, 0.15 second is required to synthesize 1 second of motion. The more example motion clips, the more time is required, but the predicted motions become more accurate. We notice that the variation of the synthesized motions relies on the variation of example motion clips in the database, but we are still unclear about how the size of database affects the naturalness of synthesized motions. Therefore, a reasonable scheme for filtering example motion clips remains an area for future work.

Theoretically, our model can be adapted for many other motion behaviors besides locomotion, such as boxing and kicking. For these behaviors, the hitting position might be chosen as the style parameter. As mentioned in §5.3, different structures of parameter propagation networks will be employed for different behaviors. However, we cannot deal with the motion behaviors that cannot be intuitively parameterized by style parameters, e.g. dancing, because the weight functions cannot be estimated. A hybrid model of GPLVM and style-and-variation interpolation would be a possible solution to this problem.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. We also thank Haibin Ren for meshing skeletons and the Carnegie Mellon Graphics Lab for providing motion capture data. This paper was supported in part by the National Key Technology Research and Development Program of China, No. 2009BAK43B38 and the National Natural Science Foundation of China, No. U0935003 and No. 60970086.

References

[BC89] BRUDERLIN A., CALVERT T. W.: Goal-directed, dynamic animation of human walking. In *SIGGRAPH '89: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (1989), ACM, pp. 233–242. 3

[BSH99] BODENHEIMER B., SHLEYFMAN A. V., HODGINS J. K.: The effects of noise on the perception of animated human running. In *Computer Animation and Simulation* (1999). 2

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), ACM, pp. 97–104. 6

[Cre93] CRESSIE N.: *Statistics for spatial data*. Wiley, 1993. 5

[GBT04] GLARDON P., BOULIC R., THALMANN D.: A coherent locomotion engine extrapolating beyond experimental data. In *Proceedings of Computer Animation and Social Agents* (2004), pp. 73–84. 2

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 522–531. 2

[HM71] HUIJBREGTS C., MATHERON G.: Universal kriging. In *Proceedings of International Symposium on Techniques for Decision-Making in Mineral Industry* (1971), pp. 159–169. 1, 6

[HPP05] HSU E., PULLI K., POPOVIĆ J.: Style translation for human motion. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 1082–1089. 3

[HW98] HARRIS C. M., WOLPERT D. M.: Signal-dependent noise determines motor planning. *Nature 394* (1998), 780–784. 2

[IF04] IKEMOTO L., FORSYTH D. A.: Enriching a motion collection by transplanting limbs. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 99–108. 3

[IST01] IK SOO L., THALMANN D.: Key-posture extraction out of human motion data. In *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.01CH37272)* (2001). 4

[JLLL08] JANG W.-S., LEE W.-K., LEE I.-K., LEE J.: Enriching a motion database by analogous combination of partial human motions. *The Visual Computer: International Journal of Computer Graphics 24*, 4 (2008), 271–280. 3, 4

[JW02] JI P., WU H.: An efficient approach to the forward kinematics of a planar parallel manipulator with similar platforms. *IEEE Transactions on Robotics 18*, 4 (2002), 647–649. 3

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 559–568. 2

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics (TOG) 21*, 3 (2002), 473–482. 6

[KSG02] KOVAR L., SCHREINER J., GLEICHER M.: Footskate cleanup for motion capture editing. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), ACM, pp. 97–104. 5

[Law03] LAWRENCE N. D.: Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems (NIPS)* (2003). 2, 8

[LBJK09] LAU M., BAR-JOSEPH Z., KUFFNER J.: Modeling spatial and temporal variation in motion data. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 1–10. 2, 5, 8

[LM07] LAWRENCE N. D., MOORE A. J.: Hierarchical Gaussian process latent variable models. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning* (2007), ACM, pp. 481–488. 2

[MK05] MUKAI T., KURIYAMA S.: Geostatistical motion interpolation. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 1062–1070. 2, 6, 8

[MLD*08] MCDONNELL R., LARKIN M., DOBBYN S., COLLINS S., O'SULLIVAN C.: Clone attack! Perception of crowd variety. *ACM Transactions on Graphics (TOG) 27*, 3 (2008), 1–8. 1

[PB00] PULLEN K., BREGLER C.: Animating by multi-level sampling. *Computer Animation* (2000), 36. 2

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: texturing and synthesis. *ACM Transactions on Graphics (TOG) 21*, 3 (2002), 501–508. 2

[Per95] PERLIN K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (1995), 5–15. 2, 8

[PSS02] PARK S. I., SHIN H. J., SHIN S. Y.: On-line locomotion generation based on motion blending. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), ACM, pp. 105–111. 3, 4

[RBC98] ROSE C., BODENHEIMER B., COHEN M. F.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18* (1998), 32–40. 2, 8

[TLP07] TREUILLE A., LEE Y., POPOVIĆ Z.: Near-optimal character animation with continuous control. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 7. 6

[UGB*04] URTASUN R., GLARDON P., BOULIC R., THALMANN D., FUA P.: Style-based motion synthesis. *Computer Graphics Forum 23*, 4 (2004), 1–14. 2

[WFH07] WANG J. M., FLEET D. J., HERTZMANN A.: Multifactor gaussian process models for style-content separation. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning* (2007), ACM, pp. 975–982. 2