Human Motion Synthesis by Motion Manifold Learning and Motion Primitive Segmentation

Chan-Su Lee and Ahmed Elgammal

Rutgers University, Piscataway, NJ, USA
{chansu, elgammal}@cs.rutgers.edu

Abstract. We propose motion manifold learning and motion primitive segmentation framework for human motion synthesis from motion-captured data. High dimensional motion capture date are represented using a low dimensional representation by topology preserving network, which maps similar motion instances to the neighborhood points on the low dimensional motion manifold. Nonlinear manifold learning between a low dimensional manifold representation and high dimensional motion data provides a generative model to synthesize new motion sequence by controlling trajectory on the low dimensional motion manifold. We segment motion primitives by analyzing low dimensional representation of body poses through motion from motion captured data. Clustering techniques like k-means algorithms are used to find motion primitives after dimensionality reduction. Motion dynamics in training sequences can be described by transition characteristics of motion primitives. The transition matrix represents the temporal dynamics of the motion with Markovian assumption. We can generate new motion sequences by perturbing the temporal dynamics.

1 Introductions

In this paper, we present a framework to synthesize human motion by combining motion primitives. Biological study shows that complicated human motions are controlled by linear combination of computational motion primitives called force fields [10]. We learn a generative model with a low dimensional motion manifold representation similar to force fields of motion primitives. To model smooth variations in human motions according to force fields, we learn nonlinear mapping between motion manifold representation and high dimensional motion data. We also model continuous human motion dynamics by sequences of primitive motions.

A low dimensional manifold representation of high dimensional human motion data provides a compact representation for analysis of human motion sequences. It also provides means to control human motion in the low dimensional space after learning a mapping between the low dimensional manifold points and high dimensional motion capture data. We use self organizing maps (SOMs) as a topology preserving network. Using SOMs, we can represent high dimensional human motion data into low dimensional Euclidean space preserving neighborhood relationship. By learning nonlinear mappings between low dimensional manifold points and high dimensional motion capture data, we can generate new motion sequences according to trajectories on the low dimensional motion manifold.

F.J. Perales and R.B. Fisher (Eds.): AMDO 2006, LNCS 4069, pp. 464–473, 2006.

We segment a given sequence of motion into sub-motion primitive by utilizing low dimensional representation of human motion sequence and clustering in the low dimensional space. There are several works related to macro-level motion segmentation, where the motion is segmented into higher level meaningful categories like walk, run, jump and so on. However, we need to find micro-level motion patterns in order to describe simple motion by the combination of the sub-motions. It is not obvious how to define the sub-motion. Recently, huge motion capture data are available in public. Therefore, we find sub-motion primitives by analyzing large motion capture data set. Dimensionality reduction techniques are applied followed by applying clustering to find sub-motion primitive in order to represent intrinsic characteristics of motion efficiently.

To model temporal dynamics of a given motion sequence and to be able to generate new motion sequences that fit to the original motion dynamics, we model motion dynamics by the transition characteristics of sub-motion primitive. Motion dynamics can be captured using transition probabilities from one primitive motion to another primitive transition after segmenting whole sequence of motion into sub-motion primitives. With Markovian assumption, we model the motion dynamics characteristics in a transition matrix of motion primitives.

2 Related Work

Machine-learning techniques are used in increasing number of papers in computer graphics, especially in data-driven motion synthesis. A stylistic hidden Markov model (SHMM), which is an HMM whose parameters are functionally controlled by a style parameter, was used for stylistic motion synthesis [4]. Scaled Gaussian Process Latent Variable Model (SGPLVM) was used to solve inverse kinematics system based on a learned model [8].

There are several different approaches to segment continuous motion sequences. One of the well-known approaches in computer vision is using hidden Markov model (HMM) [5]. Statistical approaches like Principal Component Analysis (PCA), Probabilistic PCA and Gaussian mixture model (GMM), are used to segment motion capture data into distinct behavior segment [1]. Recently there are approaches to use sub-motion sequences for segmentation. Bettinger and Cootes [2] modeled facial motion by segmenting sub-trajectories, grouping similar sub-trajectories and learning temporal relations between groups in order to model facial behavior. Temporal relationship between groups was modeled by variable length Markov model [7]. New sequence can be generated by transition of group from the learned model and sampling principal component in subgroup to find new shape of motion. For the interpolation of two sub-motion, linear model is used to avoid perceptible jumps in the generated video. Clustering techniques are also used to find key-frame in motion analysis [3].

In this paper, we employed also clustering technique similar to [3] to discover motion primitive. However, we use low dimensional motion manifold for the representation of dynamic human motion in low dimensional space, which allows low dimensional representation of high dimensional data. In addition, we learn a nonlinear generative model to synthesize details of the original motions in spite of the low dimensional representation.

3 Learning Low Dimensional Motion Manifold

We represent high dimensional human motion using a low dimensional embedded manifold representation. Then, We learn nonlinear mapping between the low dimensional manifold representation and the original high dimensional motion. The low dimensional manifold representation is motivated by force fields in the biological study of human motion [10]. The motion primitives that we are interested in are relevant to the intrinsic body configuration and irrelevant to the position and orientation of the body. In the preprocessing, we normalize body location and orientation. Now, we can represent body configuration by 3D locations of body joint instead of joint angles. This allows coordinate invariant similarity measure for body pose [9], which may be close to human perception. If we use joint angle, we need to count hierarchy of joint angle in comparison as the small difference of joint angle in higher level can cause large difference of joint location than the same amount of difference in lower level joint angle. Two motion capture datasets are used in the experiments. One is ballet motion and the other is normal walking motion.

3.1 Low-Dimensional Manifold Representation of Human Motion

We applied two manifold learning techniques for motion captured data to find low dimensional manifold representation of motion sequences. First, we find low dimensional representation of each body pose by applying Principal Component Analysis (PCA) using singular value decomposition (SVD). With the first few PCs, we can distinguish each frames with similarity relations.

Second, we applied Kohonen's self organizing map. Kohonen's neural network model was motivated by neurophysiology. The neuron layer acts as a *topographic feature map*, if the location of the most strongly excited neurons is correlated in a regular and continuous fashion with a restricted number of signal features of interest. Neighboring excited locations in the layer then correspond to stimuli with similar features [13]. Figure 1 shows two dimensional representation for walking sequence and ballet motion sequence. We can notice that the representation points spread in all the space (Figure 1 (b)). In Figure 1 (a), We can notice three cycling patterns through the path. However, in SOM, even the similar motion cycles are represented in different locations and are spread in the space. You can see similar patterns in Figure 1 (c) (d), which is the case of complicated ballet motion.

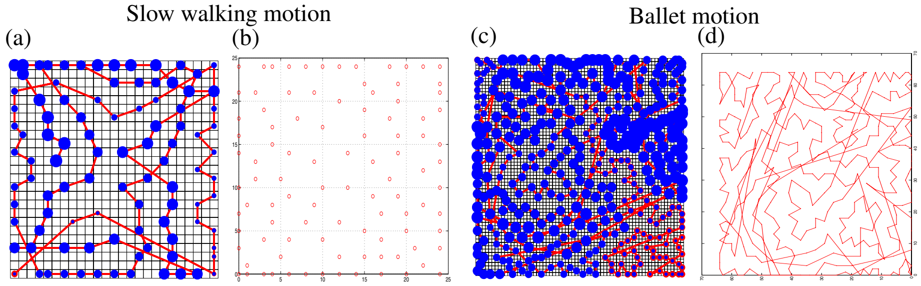


Fig. 1. SOM analysis for simple walking (a) (b) and complicated ballet motion (c) (d)

3.2 Learning Generative Models Using Motion Manifold

We learn nonlinear mapping between the manifold embedding and original motion in order to generate new motions based on embedded manifold points. Suppose that we can learn a nonlinearly embedded representation of the high dimensional motion manifold M in a low dimensional Euclidean embedding space, R^e, then we can learn a set of mapping functions from the embedding space into the input space, i.e., functions $\gamma(x_t) : R^e \rightarrow R^d$ that maps from embedding space with dimensionality e into the input space (observation) with dimensionality d. Since the embedding and the original data are related by nonlinear manifold learning, we need to learn nonlinear mapping in order to capture motion characteristics accurately. In particular we consider nonlinear mapping functions of the form

$$y_t = \gamma(x_t) = B \cdot \psi(x_t) \tag{1}$$

where B is a $d \times N$ linear mapping and $\psi(\cdot) : R^e \rightarrow R^N$ is a nonlinear mapping where N radial basis functions can be used to model the manifold in the embedding space, i.e.,

$$\psi(\cdot) = [\psi_1(\cdot), \cdots, \psi_N(\cdot)]^T$$

For i-th frame y_i, which is sampled data of y_t at time $t = i \cdot \frac{N}{T}$, we can find low dimensional embedding point X_i. Given an embedded manifold representation $x_i, i = 1 \cdots N$ in e dimensional embedding space for $y_i, i = 1 \cdots N$, we can learn nonlinear mappings $f : R^e \rightarrow R^d$ using generalized radial basis function (GRBF) interpolation [12] to the original sequence y_t by solving for multiple interpolants, i.e., $f^l : R^e \rightarrow R$ for each tracking feature l. We can use thin-plate spline $(\phi(u) = u^2 log(u))$ or Gaussian $(\phi(u) = exp(u))$ as the basis function. The whole mapping for sequence k can be written in a matrix form as

$$f_k(x) = B^k \cdot \psi(x) \tag{2}$$

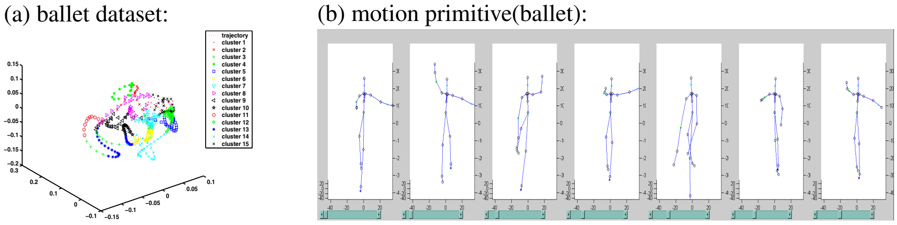
where B^k is a coefficient for the generative model of motion data.

4 Motion Primitive Segmentation and Motion Dynamics Modeling

We segment primitive motions from the low dimensional manifold representation. Based on segmented motion primitive, we can model dynamics of human motion by transition probability of motion primitives.

4.1 Finding Primitive Motion Using Clustering

The representative motion primitive is estimated by clustering of the low dimensional representation of motion sequence. At first, we applied standard k-means algorithm and measured error in a given k clusters. We estimate the natural number of primitive by estimating error in different number of clusters and finding elbow in the error graph for different number of clusters. Based on the reconstruction error according to the number of cluster, we can decide the number of clusters. In our data set, we find that the ballet


(a) ballet dataset:

(b) motion primitive(ballet):

Fig. 2. Clustering motion sequences

motion shows 15 clusters and the walking sequence shows 10 clusters in the estimation of natural number of clusters. After finding natural number of cluster, we applied fuzzy k-means algorithm and Gaussian mixture model clustering using estimated natural cluster number. Fuzzy k-means clustering result shows better clustering result with respect to the inner distance within cluster and separation between clusters. Figure 2 shows clustering result by fuzzy k-means algorithms for ballet motion with 10 clusters (a). Figure 2 (b) shows body poses corresponding to the centers of the first seven clusters in ballet motion dataset. In order to find proper sequence of each cluster for continuous motion generation, we need to model dynamics of the motions.

4.2 Modeling Temporal Dynamics Using Markov Chains

Temporal dynamics of the motions are modeled using Markov chains. A Markov assumption assumes that the next state of a system (S_{t+1}) is only dependent on the previous n states (S_t, S_{t-1}, S_{t-2}, $\cdots$, S_{t-n+1}). By assuming that transition to new motion primitive (new state) depends only on current motion primitive class (current state), we modeled motion dynamics as a first order Markov model. Now, the likelihood of one primitive cluster following another can be expressed as a conditional probability $P(S_{t+1}|S_t)$. Transition probability from state j at time t to state k at time $t+1$

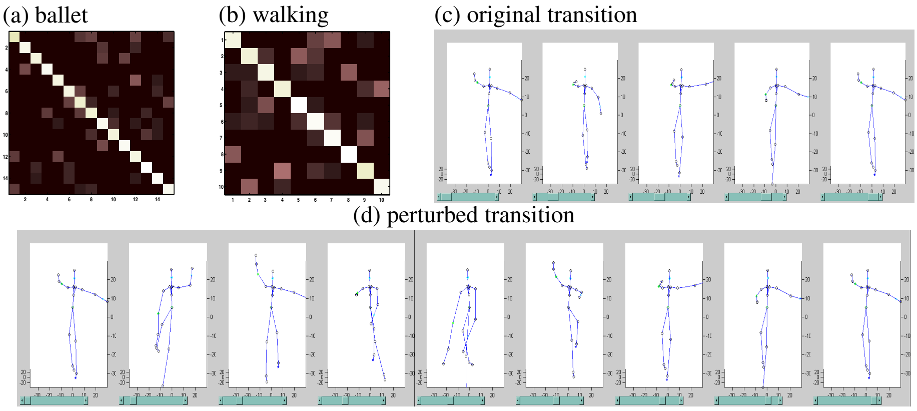
$$p_{k,j} = P(C_k^{t+1}|C_j^t), \tag{3}$$

where $P(C_j^t)$ denotes the unconditional probability of being in cluster j at time t, can be estimated easily by counting two adjacent frames cluster transition in the original data set.

A transition matrix can model the whole dynamics

$$\begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots, \\ p_{n,1} & \cdots & p_{n,n} \end{pmatrix} \tag{4}$$

where $\sum_j p_{k,j} = 1$ for all j, and n is the number of clusters in the model. Figure 3 shows transition matrices for ballet (a) and walking (b) datasets. The bright color means high probability of transition. The figure show highest probability in the diagonal, which means most likely next frame is within the same cluster. We can estimate most likely



(a) ballet (b) walking (c) original transition

(d) perturbed transition

Fig. 3. Transition matrices and transition of motion states

next primitive motion cluster k^* by choosing the next highest probabilistic transition from cluster j.

$$k^* = \arg\max_i p_{i,j},\ i \neq j \tag{5}$$

in the transition matrix. Figure 3 (c) shows motion transition sequence estimated by the most second likely transition state from one selected primitive motion until it return back to the state. We can get new motion transition sequence by perturbing transition matrix with small noise as shown figure 3 (d).

5 Synthesis of Human Motion Using Motion Manifold and Motion Primitive

We can synthesize a new motion sequence in two ways. First, we can directly synthesize new motion sequence from any low dimensional trajectory since we can generate motion sequences for any given manifold points given the learned nonlinear generative model. Second, we can generate dynamic sequences of motion based on the transition model which is learned from training sequence.

5.1 Direct Motion Synthesis Using Low Dimensional Motion Manifold

We implemented low dimensional representation of ballet motion using SOM. First we learn SOM by 65×65 lattice structure (Actually, we tried smaller number of lattice such as 25×25, 40×40 or 50×50. In these case, some motion fired in the same lattice location, which is not good for learning as the same low dimensional representation point requires learning to reconstruct two different high dimensional data). After finding different lattice representation, we used small number of regular lattice center as the basis center for radial basis function. We used 15×15 number of radial bases for GRBF learning. After that we implemented two kinds of interaction methods: manifold point based synthesis and given key motion based synthesis.

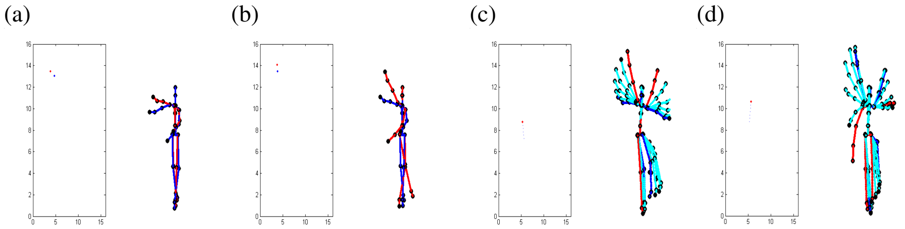


Fig. 4. Motion synthesis: (a) (b) Point interaction in low dimensional space (c) (d) Path interpolation in low dimensional space

In the manifold point-based approach, user selects points on the manifold using mouse. After finding the location of the mouse click point within the given manifold, we can generate motion based on trajectory of selected points. Figure 4 (a) (b) shows last selected point (blue) and newly selected point (red) and their corresponding reconstructed motion. It shows continuous variation of the motion when we interpolate points on the manifold and generate intermediate motion corresponding to intermediate manifold points. When multiple points are selected, we do spline fitting for the selected manifold points for smooth interpolation of intermediate motion. Figure 4 (c) (d) shows examples of the interpolating intermediate motion. Blue color motion is the motion corresponding to the last mouse click. Red color represent new mouse click location. Intermediate motions are generated as shown in the figure (cyan color).

The other method is based on given key motions. Using inverse mapping, we can find a low dimensional representation for a given new key motion. In the case of SOM,

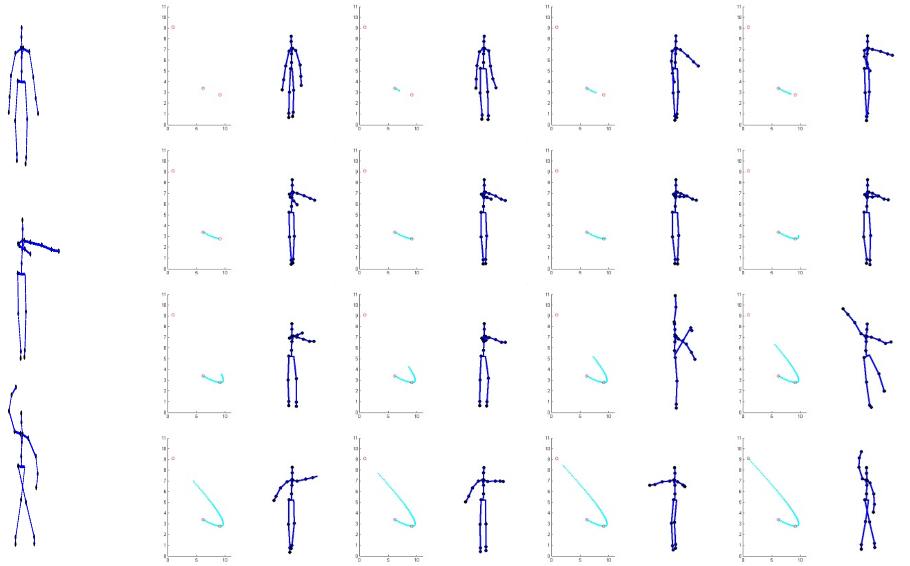
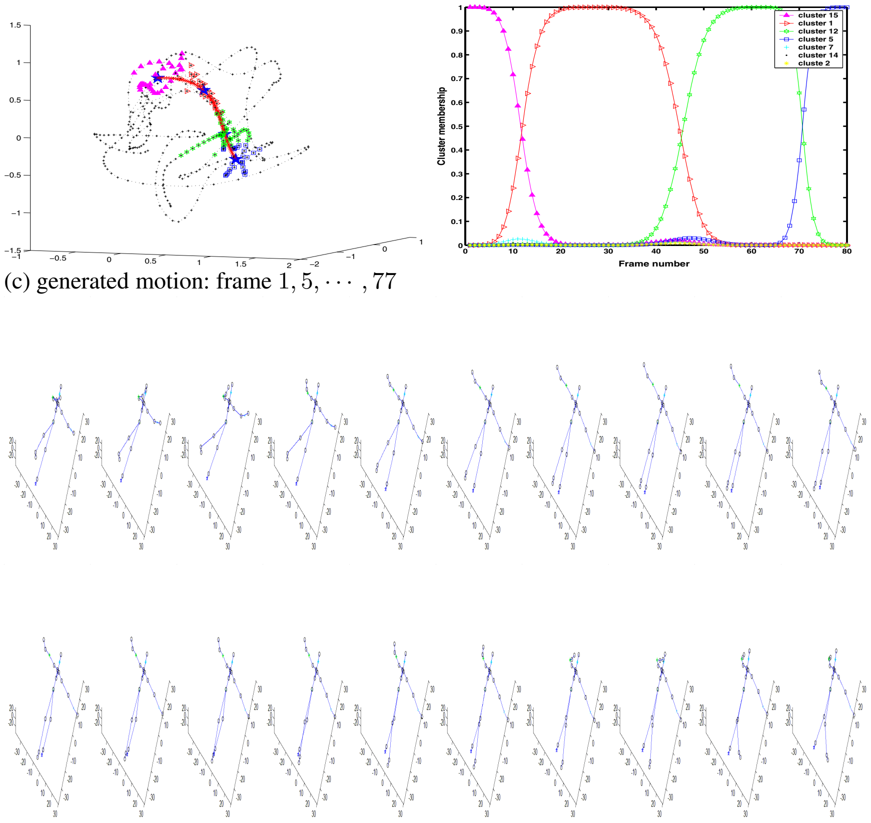


Fig. 5. Path interpolation in low dimensional space

we can find low dimensional manifold representation for given motion frame by finding *Best Matching Unit (BMU)* in the original lattice and scale it to the mapping coordinate space. In other case, we can achieve approximate solution using polynomial terms of GRBF [12].

Figure 5 shows an example of motion synthesis based on given key motions. In the left column, three selected key motions are given. The seletect key motions are the motion we want to generate; we want to generate motion begins from the first motion and then generate second motion in the intermediate frame. Finally the animation needs to be finished in the third key motion. In the right column, we shows low dimensional manifold points and corresponding motion generated. Red markers on the motion manifold represent low dimensional location of the three sample key motions. After spline fitting, we re-sampled the spline curve for a given sample number. As we follow mapping trajectory in the low dimensional space, it shows not just interpolation of three sample points but smooth synthesis of intermediate motions based on training data. The figure shows that there are additional intermediate sub-motions in the synthesis of new motions based on given key motions.



(a) interpolation trajectory:

(b) cluster membership:

(c) generated motion: frame $1, 5, \cdots, 77$

Fig. 6. An example of motion primitive interpolation

5.2 Generation of Continuous Motion Sequence

We can generate new motion sequence for any given initial motion frame with dynamics of original motion. After finding transition sequence for given motion frame, we can define trajectory on the motion manifold by connecting sequence of motion manifold points corresponding to the given motion primitives. The deviation from the original motion sequence can be controlled by the scale factor in the perturbation of transition matrix by superimpose random noise all the transition matrix elements. We find smooth trajectory from the motion primitive sequence by spline fitting of cluster center of each corresponding motion primitives. By sampling points on the manifold points along the spline, we can generate new sequence of motions. Figure 6 shows a generated motion sequence with spline interpolation trajectory and clustering membership in each sampling point along the interpolation trajectory. Possible transition sequence was found from transition matrix and 80 points are resampled after spline fitting to the primitive centers. It shows smooth motion transitions in frame $1, 5, 9, 13, \cdots$, 77. For any given initial pose, we can generate most feasible primitive pose sequence from transition matrix with no perturbation. Figure 7 shows most likely key pose sequence when we start from two different motion frame.

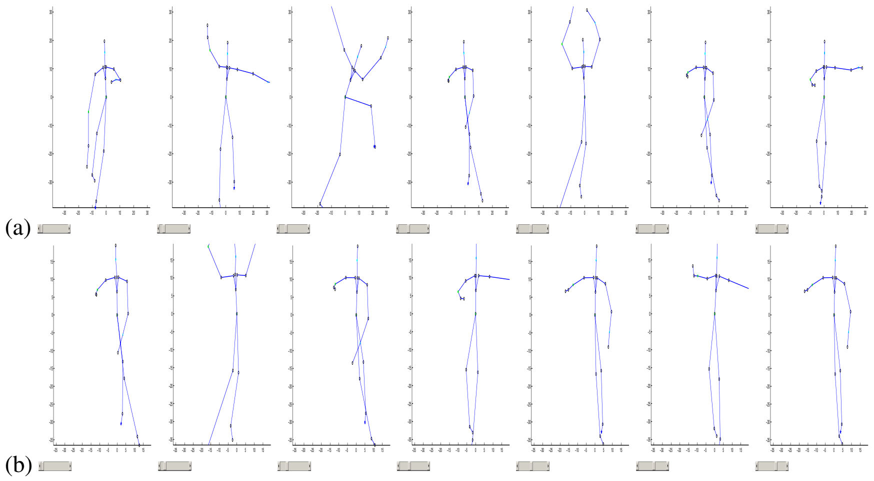


Fig. 7. Generations of following motion for given initial motion frames

6 Conclusions and Future Works

We presented an approach to generate new motion sequences using statistical analysis and learning techniques. This approach is more flexible and close to human motion generation mechanism as it generates sequence of motion based on motion primitive and transition probabilities among motion primitives. Motion primitives found by clustering of given data set is somewhat dependant on the given data set and the number of clusters, even though we find natural number of cluster for the given data set, which may compensate for the dependence of motion primitive to the given data set. However, this motion primitives can summarize whole motion sequence with small motion primitives and it simplifies representation and transition model and makes the problem solvable with simple model. The framework presented in this paper can be applicable in motion analysis in computer vision problem. It will be elegant to combine video data with motion capture data: tracking and recognizing human motion from video sequences with possible motion sequence representation from motion capture data.

For more complicated and general motion primitives, we may need to count hierarchical representation of motion primitive as in [11]. Modeling transition of sub-motion is simplified assuming the first-order Markovian dynamics, which may not enough to capture complicated motion transitions. We may use more rich representation like variable length Markov model [7] or higher order Markov models. We can extend the generative models to cover variations in different person as style factors similar to [6].

References

1. J. Barbic, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proc. of Graphics Interface*, 2004.
2. F. Bettinger and T. F. Cootes. A model of facial behaviour. In *Proc. of FGR*, pages 123–128, 2004.
3. R. Bowden. Learning statistical models of human motion. In *Proc. of IEEE Workshop on Human Modeling, Analysis & Synthesis*, 2000.
4. M. Brand and A. Hertzmann. Style machines. In *Proc. of SIGGRAPH*, pages 183–192, 2000.
5. M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. on PAMI*, 22(8), 2000.
6. A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proc. CVPR*, volume 1, pages 478–485, 2004.
7. A. Galata, N. Johnson, and D. Hogg. Learning variable-length markov models of behavior. *Computer Vision and Image Understanding*, 81:398–413, 2001.
8. K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004.
9. L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *Proc. of SCA*, pages 214 – 224, 2003.
10. F. Mussa-Ivaldi and E. Bizzi. Motor learning through the combination of primitives. *Philosopical Transactions of the Royal Society of London Seris B, Biological Science*, 355:1755–1769, 2000.
11. S. Park and J. K. Aggarwal. Recognition of two-person interactions using a hierarchical bayesian network. In *Proc. of Workshop on Video surveillance*, pages 65–76. ACM Press, 2003.
12. T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. IEEE*, 78(9):1481–1497, 1990.
13. H. Ritter, T. Martinetz, and K. Schulten. *Nueral Computation and Self-Organizing Maps*. Addison-Wesley, 1991.