Simultaneous Particle Tracking in Multi-Action Motion Models with Synthesized Paths

Norimichi Ukita

Nara Institute of Science and Technology, Ikoma, Nara, 630-0192 Japan

Abstract

This paper proposes human motion models of multiple actions for 3D pose tracking. A training pose sequence of each action, such as walking and jogging, is separately recorded by a motion capture system and modeled independently. This independent modeling of action-specific motions allows us 1) to optimize each model in accordance with only its respective motion and 2) to improve the scalability of the models. Unlike existing approaches with similar motion models (e.g. switching dynamical models), our pose tracking method uses the multiple models simultaneously for coping with ambiguous motions. For robust tracking with the multiple models, particle filtering is employed so that particles are distributed simultaneously in the models. Efficient use of the particles can be achieved by locating many particles in the model corresponding to an action that is currently observed. For transferring the particles among the models in quick response to changes in the action, transition paths are synthesized between the different models in order to virtually prepare inter-action motions. Experimental results demonstrate that the proposed models improve accuracy in pose tracking.

Keywords: Human pose tracking, Motion prior, Multiple actions, Action transition, Transition paths

1. Introduction

To estimate the complex poses of a human body in videos (see survey articles [1, 2, 3]), pose tracking using a motion prior is effective. The motion prior is useful for resolving short-lasting ambiguities between a body pose and its image features. The more detailed and precise the prior becomes, the more accurate the estimation result is. The precise prior of the human body can be obtained by a motion capture system. Several kinds of actions, such as walking and running, are recorded in motion datasets that are widely used for modeling and evaluating human motions in Computer Vision [4] and Graphics [5, 11] communities. The motion model of each action can be used for pose tracking in that action.

A set of motion models for multiple actions is proposed in this paper. In addition to elemental actions recorded in a motion dataset, transitions among these actions (e.g. from walking to jogging) are also observed in natural scenarios. Potential transitions among all of the actions are extremely varied (i.e. including intra-individual and inter-individual variations). Recording all of these variations is unrealistic. If the motion priors of the multiple actions are modeled with no transitions, pose tracking over different actions is difficult.

This paper proposes a set of motion models that explicitly represent the smooth transitions among elemental actions, which are recorded in the dataset. In the proposed motion modeling, the smooth transitions are synthesized from the original dataset and used for modeling a motion prior among different actions. In pose tracking with the proposed models, unlike switching dynamical models, the multiple motion models are employed simultaneously for improving robustness of motion prediction; even if a motion model fails to predict the motion of a target person, another model would be able to follow that motion.

After introducing related work (Sec. 2) and previous methods for pose tracking with a single action model (Sec. 3), Sec. 4 describes how to build the proposed models with multiple actions and how to use the models for pose tracking. Experimental results with the proposed models are presented in Sec. 6, and we conclude the paper in Sec. 7.

2. Related Work

Human pose estimation is mainly classified into model fitting [6] and feature-to-pose regression [7]. Model fitting is achieved by adjusting a pose of a skeletal model (i.e. a set of joint angles/positions) so that the pose fits into the image features of a human body (e.g. edge, silhouette, volume). In pose regression, a regression function between synchronized poses and image features is employed. The function is obtained in advance from real images and pose data that are captured simultaneously by a camera(s) and a motion capture system.

Both for model fitting and pose regression, temporal pose tracking [8] is superior than pose estimation at each frame. For tracking robust to failure in image processing, particle filtering is effective; see [40], for example. While particle filtering is successful for many visual tracking problems because of its robustness and ability to recover from tracking failure, applying it to human pose tracking is not easy due to two reasons: curse of dimensionality and complex body-motion.

Curse of dimensionality: Although particle filtering can be done in a relatively high-dimensional space by using particle reposition [9] and coarse-to-fine processing [10] (e.g. annealed

particle filtering [46]), it is difficult in a more high-dimensional space (30-D or more) such as a human pose space.

Complex body-motion: In human pose tracking, a motion model is needed for predicting a pose at $t+1$ from that at t. Human motion has been modeled by various ways: interpolation [12], Gaussian mixture models [13, 14], Hidden Markov Models (HMM) [16], Variable Length Markov Model [17], manifold [18], exemplar (retrieval) model [19], autoregressive model [20], and Relevant Vector Machine (RVM) regression [21]. Generalization and accuracy of a motion model are crucial for correct pose tracking. Indeed, high dimensionality of the joint angles/positions (i.e. 30–60 dimensions) and their erratic motions make it difficult to represent various motions efficiently and correctly.

In most of the motion models, high-dimensional complex and erratic motions are modeled probabilistically (e.g. by using HMM or Gaussian) in lower dimension (e.g. by using PCA, Isomap [22], LLE [23], LTSA [24], or their combinations [25]). In such a low-dimensional space, particle filtering works well. For low-dimensional modeling, nonlinear probabilistic embedding such as Gaussian Process Latent Variable Models (GPLVMs [26]) is widely used recently. GPLVM is superior to the above mentioned dimensionality reduction methods in terms of its abilities of nonlinear mapping and stochastic modeling. Compared with linear mapping, nonlinear mapping can precisely express the complex mapping between higher-dimensional and lower-dimensional spaces. The stochastic representation tells us which pose is often observed; this information is useful for determining the likelihood of each observation in a pose tracking process.

Several extensions of GPLVM have been studied, for example, dynamics representation [35], bidirectional smooth mapping between latent and observation spaces [27], hierarchical representation [28], and a shared latent structure that allows us to connect multiple observation spaces [37].

Above all, Gaussian Process Dynamical Model (GPDM) [35] is useful for modeling temporal data such as human motion. This is because GPDM provides a low-dimensional model and a motion prediction function in the model simultaneously. The motion prediction function in the low-dimensional space allows us to resolve the above mentioned two problems in particle filtering for human pose tracking (i.e. high-dimensionality of a human body pose and its complex motions).

These latent models with Gaussian Process (GP) allow us to model multiple kinds of actions as well as a single action; a set of independently trained models [29, 30] and a unified model with multiple actions trained together [29, 47, 31, 32]. These multi-action models allow us to achieve pose tracking in a general long sequence, in which different actions are sequentially observed. While selection of an appropriate model at each moment is required for a set of the independent models, it has advantages with respect to scalability and accuracy; 1) since the computational cost of modeling grows as sample data increases in each model, (i.e. the computational cost is expressed by $O(N_S^3)$ [26] where N_S is the number of samples[1]), the independent models allow us to compute each model fast and 2) each model is optimized for its respective action. However, neither independent nor unified models are good at estimating motion transitions between different actions if such motion transitions are not included in sample data for training. As mentioned in the introduction, recording all of possible transitions among a number of actions is difficult.

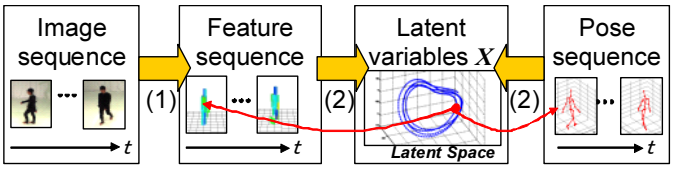


Figure 1: Learning the motion prior and feature-to-pose regression of a single action. (1) Feature extraction (e.g. shape contexts [43]). (2) Shared latent-structure modeling with GPDM. Each latent variable corresponds to its respective feature and pose as depicted by red arrows.

Synthesizing realistic transitions between different poses has been widely studied in Computer Graphics. For synthesizing transitions among arbitrary motion sequences, motion graphs [34] are useful. The goal of motion graphs is to synthesize a new sequence as visually natural as possible. In motion graphs, new transitions are synthesized by connecting (i.e. interpolating) sample sequences via similar poses in different sequences.

We propose 1) how to integrate the advantages of GP latent models for a low-dimensional principal representation and motion graphs for synthesizing a variety of realistic transitions among actions in the high-dimensional observation spaces and 2) how to achieve accurate pose tracking in the models.

3. Pose Tracking in a Single Action: Previous Works

The proposed pose tracking is based on pose regression from image features. This section describes previous methods for pose tracking of a single action. Its learning scheme is described in Sec. 3.1 and 3.2, while its tracking scheme is explained in Sec. 3.3. The process flows of the learning and tracking schemes are illustrated in Fig. 1 and 3, respectively.

3.1. Motion Modeling by Gaussian Process Dynamical Models

Gaussian Process Dynamical Models (GPDM) [35] (Fig. 2) represent the smooth dynamics of sample data in a low-dimensional latent space X (e.g. "Pose latent space" in Fig. 2) from high-dimensional observed data in an observation space Y, (e.g. "Pose observation space" in Fig. 2), namely joint angles in our experiments. Inherently a GP allows us to generalize the latent space increasing its conformity with human body kinematics. GPDM is defined by two mapping functions; 1) from a point at t to a point at $t+1$ in the latent space, $\boldsymbol{x}_{t+1} = f_D(\boldsymbol{x}_t)$ where $\boldsymbol{x}_t, \boldsymbol{x}_{t+1} \in X$, and 2) from the latent space to the observation space, $\boldsymbol{y}_t = f_O(\boldsymbol{x}_t)$, where $\boldsymbol{y}_t \in Y$. $f_D(\boldsymbol{x}_t)$ gives us the capability of temporal prediction that is useful for tracking. (e.g.

[1] Even a fast stochastic gradient-descent approach [33] slows as sample data increase.

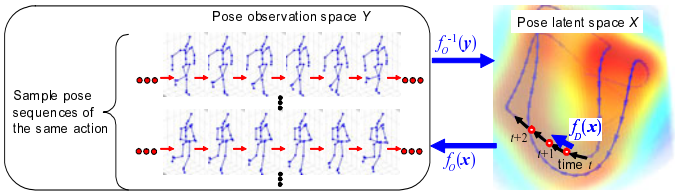


Figure 2: GPDM [35] provides 1) a mapping function from a latent space X to its observation space Y and 2) a temporal mapping function from a latent variable at $t-1$ to that at t. Circles and arrows in X depict latent variables and temporal mapping, respectively. The background color in X denotes the variance at each point; lower variance (red) to higher variance (blue).

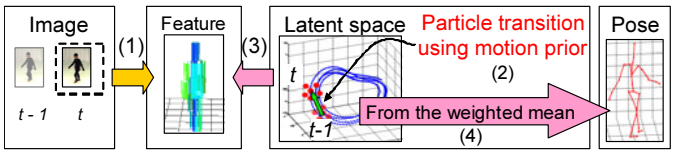


Figure 3: Pose tracking with a motion model. (1) Feature extraction. (2) Particle tracking using a motion prior. (3) Mapping the particles into the feature space for evaluating its likelihood. (4) Mapping the likelihood-weighted mean of the particles to the pose space.

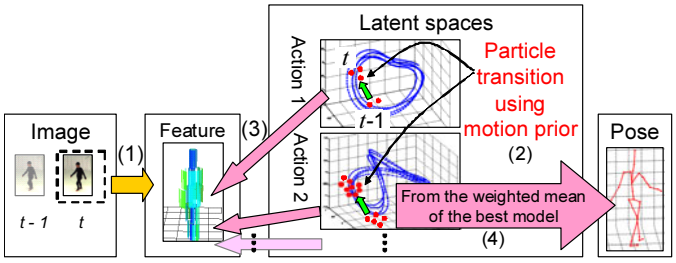


Figure 4: Pose tracking with multiple motion models. (1), (2), and (3) are same with those in Fig. 3, except that particles are propagated in the multiple models simultaneously. (4) Mapping the likelihood-weighted sum of the particles from the latent space that has the maximum likelihood into the pose space.

Bayesian tracking [8] and particle filtering [38]). The nature of GP also provides the distribution (variance) of data. More specifically, the variance of $\boldsymbol{x}_t$ is calculated.

Given a sample sequence of high-dimensional observation data $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N]$, where N denotes the number of frames, the mapping functions are acquired by maximizing the joint likelihood of $\boldsymbol{Y}$ and $\boldsymbol{X}_{2:N}$ with respect to $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ and $\boldsymbol{X}_{1:N-1}$, respectively, where $\boldsymbol{X}_{2:N} = [\boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ and $\boldsymbol{X}_{1:N-1} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N-1}]$. In this optimization, similarity between different components in $\boldsymbol{X}$ is evaluated by a nonlinear kernel function. The nonlinear kernel function was implemented by the Gaussian radial basis function in our experiments.

3.2. Shared Latent Structure for Connecting Features and Poses

For feature-to-pose regression in the proposed method, synchronized features and poses ("Feature sequence" and "Pose sequence" in Fig. 1) are modeled together. The connection between them is established by the latent space shared by their observation spaces, as shown by (2) in Fig. 1, as with pose tracking proposed in [36]. Shared latent structure modeling [37] connects two observation spaces via a single shared latent space by maximizing the joint likelihood of $\boldsymbol{Y}^P = [\boldsymbol{y}_1^P, \cdots \boldsymbol{y}_N^P]$ and $\boldsymbol{Y}^V = [\boldsymbol{y}_1^V, \cdots, \boldsymbol{y}_N^V]$ with respect to $\boldsymbol{X}$, where $\boldsymbol{Y}^P$ and $\boldsymbol{Y}^V$ denote N samples of synchronized poses and features, respectively. Then each latent variable $\boldsymbol{x}_t$ corresponds to its respective pose $\boldsymbol{y}_t^P$ and feature $\boldsymbol{y}_t^V$ as depicted by thin red arrows in Fig. 1.

In our framework, $\boldsymbol{X}$ is optimized from $\boldsymbol{Y}^P$ and $\boldsymbol{Y}^V$ with GPDM for obtaining temporal transition in $\boldsymbol{X}$; the joint likelihood of $p(\boldsymbol{Y}^P|\boldsymbol{X})$, $p(\boldsymbol{Y}^V|\boldsymbol{X})$, and $p(\boldsymbol{X}_{2:N}|\boldsymbol{X}_{1:N-1})$ is maximized. Initialization and optimization of the shared latent space are achieved in the same way as [36].

3.3. Particle Tracking for Feature-to-Pose Regression

With feature-to-pose mapping via their shared latent space, pose estimation at each frame can be achieved. For successful pose estimation, correct feature tracking is needed in an image sequence.

For feature tracking, particle filtering in the latent space, X, is performed. A particle at t in X transits from its respective one at the previous frame $(t-1)$ using motion prior $f_D(\cdot)$ as shown in Fig. 3 (2) and then is mapped to the feature space as shown Fig. 3 (3). At each frame t, a feature is extracted from a captured image as shown in Fig. 3 (1). Then the likelihood $c_{t,i}$ of i-th particle at t (denoted by $\boldsymbol{p}_{t,i}$) with respect to the feature at t (denoted by $\boldsymbol{f}_t$) is expressed as follows:

$$c_{t,i} = \exp\left(-w_v\sigma_{t,i}^2 - w_o\|f_O^V(\boldsymbol{p}_{t,i}) - \boldsymbol{f}_t\|^2\right), \quad (1)$$

where $\sigma_{t,i}^2$ and $f_O^V(\cdot)$ denote the variance of $\boldsymbol{p}_{t,i}$ in the model and the mapping function from X to the feature space, respectively. Weight variables w_v and w_o are determined empirically.

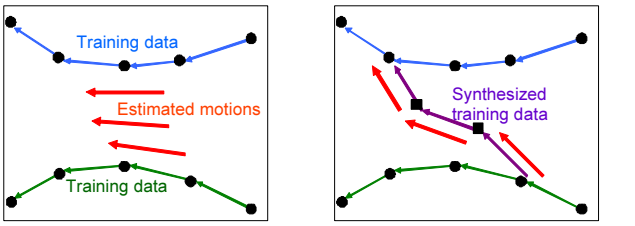
Finally, the pose is estimated by mapping the likelihood-weighted mean of the particles from X to the pose space as shown by (4) in Fig. 3.

4. Pose Tracking in Multiple Actions with Synthesized Transition Paths

4.1. Particle Tracking in the Motion Models of Multiple Actions

As mentioned in [29], independent modeling of actions can represent each action better than unified modeling in a single model. Each model is generated by the method described in Sec. 3 in the learning process.

In pose tracking with multiple action models (Fig. 4), depending on an action observed at each moment, the model corresponding to that action is selected for correct pose tracking. Model selection should take into account the history of tracking results for robustness to instantaneous observation error. While the above mentioned method [29] employs multiple action models, it has no strategy for model selection for the tracking purpose. Such model selection has been studied for tracking complicated motions; for example, switching dynamical systems [39] selects proper human motion dynamics. Switching GPDMs [30] improves tracking robustness by dimensionality



(a) No training data between two actions (b) With training data between two actions

Figure 5: Estimating motion dynamics between two different action models. Blue and green arrows depict temporal motions of original training data (i.e. body poses) depicted by black dots. Purple arrows are prepared by concatenating poses synthesized by interpolation between the original poses; the synthesized poses are illustrated by black squares. Red arrows are estimated from the motion dynamics that are modeled by training data. Red arrows in (a) are estimated only from blue and green arrows, while those in (b) are estimated from blue, green, and purple arrows.

reduction of data spaces (i.e. pose and image feature spaces) and particle filtering, where particles are propagated using the selected motion dynamics in the lower-dimensional space.

While switching GPDMs [30] combines multiple dynamical models and particle filtering in the efficient low-dimensional space, several advantages of the combination are lost:

- **Single hypothesis of motion:** In [30], particles allow us to track *multiple hypotheses of pose*, but only *one motion dynamics* is selected and applied to all the particles at each moment. If the selected motion dynamics is different from a currently observed motion, many particles are propagated to a direction different from that of the target motion.

The switching GPDMs [30] has one more problem:

- **No transition path:** In [30], it is implicitly assumed that sample motion data between multiple actions are given for learning the switching states. However, it is difficult to obtain the sample data of all possible transitions among a number of actions. If no transition path is included in motion models, quick particle transition between the models is difficult.

To resolve these two problems, the following ideas are employed in our proposed method:

- **Multiple hypotheses of motion:** Particles are distributed in multiple dynamical models that are used simultaneously for multiple hypotheses of a motion prior.

- **Synthesized transition path:** Transition paths are synthesized from the real samples of multiple actions and merged with them. The synthesized paths achieve smooth particle propagation in quick response to varying human actions; motion dynamics along the synthesized paths leads the particles to a next action model.

  While the mixture of multiple models can be used for representing the pose between different actions as proposed in [29], it might produce incorrect motion dynamics around the middle point between action models. Figure 5 (a) shows typical examples. This figure shows a feature space where motion dynamics is modeled. Blue and green arrows depict the traveling directions of training data of two different actions. Using these training data, motion dynamics around those two actions are estimated as illustrated by red arrows. All the estimated motions point to the directions same with those of the training data.

  On the other hand, our solution is to explicitly employ training data that move between those actions, which are depicted by purple arrows in Fig. 5 (b), for embedding motion dynamics between them into the models.

Specifically, pose tracking with the proposed motion models is designed in accordance with Condensation [40]:

1. Particles are randomly distributed in the total models.
2. The existing particles are drifted using motion prior $f_D(\cdot)$ at t as illustrated in (2) in Fig. 4, and then diffused so that more particles are placed near the ones having higher likelihood, which is computed by Eq. 1 at $t-1$.
3. Each particle is mapped into the feature space and compared with the feature at t for evaluating the likelihood of the particle by Eq. (1), as illustrated in (3) in Fig. 4.
4. The sum total of the likelihoods of all particles in each model is considered to be the similarity score of the model, which shows how the model fits with a currently observed action. The model having the max similarity is selected. The likelihood-weighted sum of the particles in the selected model[2] is mapped to the pose space for estimating the pose observed at t, as shown by (4) in Fig. 4.
5. If the likelihood of a particle is below a threshold, this particle is removed. Then go back to 2.

The difference from Condensation [40] is that particles are distributed in the multiple models and move among the different models by diffusion. Diffusion among the models is achieved via transition paths as well as within a model. Synthesizing the transition paths is described in Sec. 4.2. Section 4.3 describes how to diffuse the particles between the models via the synthesized transition paths. Section 4.4 shows how to maintain the scalability of the models that include the synthesized transition paths.

4.2. Synthesizing Transition Paths among Multiple Actions

In the proposed model, transition paths are synthesized from real samples and merged with them. For realistic transition, short and smooth paths are synthesized as with motion graphs [34, 41]; the path should 1) be short for synthesizing pose data that are similar to real samples by interpolation near those samples and 2) be smooth to avoid unnatural change in motion.

4.2.1. End Points of Transition Paths in Sample Data

Two end points of each transition path are determined so that their respective poses have the local maximum of a similarity

[2] One might use all particles in all models for pose regression (e.g. [29]).

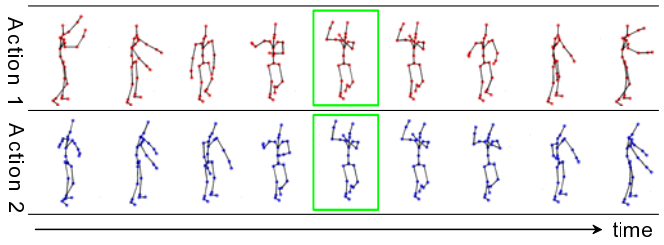


Figure 6: Local maximum of similarity (indicated by green rectangles) that correspond to the end points of transition paths. Two kinds of dance sequences were used: a subject moved the arms "right-upper and left-upper" and "right-upper and left-lower" in actions 1 and 2, respectively.

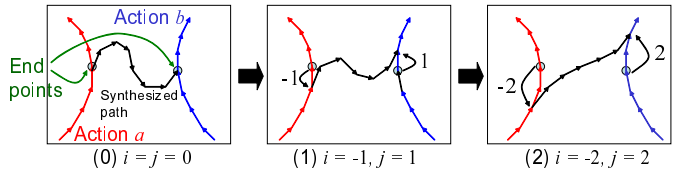


Figure 7: Interpolating pose data between the extracted end points that connect actions a to b.

score between pose vectors $\boldsymbol{y}$ of two actions a and b, $-\|\boldsymbol{y}_i^a - \boldsymbol{y}_j^b\|$, where subscripts denote i-th and j-th frames, and the similarity score is above a threshold, which was given manually. For the similarity score, $\boldsymbol{y}$ consists of 3D positions of all joints.

In an example of Fig. 6, two poses enclosed by green rectangles show a pair of the poses that have the local maximum of the similarity score.

While only one example is shown in Fig. 6, other pairs of similar poses between two actions are also extracted as the end points of transition paths, if each of the pairs has the local maximum of the similarity score. In the example of Fig. 6, the end points can be extracted when two arms are located in front of the body (e.g. the third images from the left and the second images from the right in Fig. 6). In examples of walking and jogging actions, which were also used in experiments shown in Sec. 6, the end points can be found whenever the left or right legs touched the ground. In all pairs of the end points, transition paths are synthesized as described in the Sec. 4.2.2.

4.2.2. Synthesizing Pose and Feature Data on Transition Paths

New paths are synthesized by interpolating sample poses between the extracted end points. The poses are interpolated in the joint angle representation, which are represented by quaternion, because interpolating joint positions would cause the change in the length of each limb. While naive interpolation between the samples is executed in the original motion graphs [34], good connectivity is achieved by finding the shortest path between the samples via a number of intermediate interpolation poses in [41]. These synthesized paths meet the requirement for short and smooth transitions.

In the proposed method, an additional constraint for smooth transition is employed by taking into account the smoothness of the sample motion. The existing methods [34, 41] control the smoothness by adjusting the number of interpolating points. How to determine the number of the points is important, which has not been discussed in the existing methods. Our method determines the number of the points adaptively depending on the sample motion so that the curvature of the synthesized path is less than the largest curvature of the samples that are near the transition path. Specifically, synthesizing the transition path is designed as follows:

- **P1:** Assume that end points, $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_j^b$ ("End points" in Fig. 7), are found between actions a and b. Let θ_i^A be an angle between $\boldsymbol{y}_i^A$ and $\boldsymbol{y}_{i-1}^A$, where $A \in a, b$. The angle is defined by the inner product of $\boldsymbol{y}_i^A$ and $\boldsymbol{y}_{i-1}^A$: $\cos\theta_i^A = (\boldsymbol{y}_i^A \cdot \boldsymbol{y}_{i-1}^A)/\|\boldsymbol{y}_i^A\| \cdot \|\boldsymbol{y}_{i-1}^A\|$. $\ddot{\theta}_i^A$ denotes its second derivative: $\ddot{\theta}_i^A = (\theta_i^A - \theta_{i-1}^A) - (\theta_{i-1}^A - \theta_{i-2}^A)$. The maximum $\ddot{\theta}_i^A$ (denoted by $\ddot{\theta}_{max}^A$) and $\|\boldsymbol{y}_{i+1}^A - \boldsymbol{y}_i^A\|$ (denoted by y_{max}) are computed among $i \pm t$ and $j \pm t$, where $t \in \{1, \cdots, T\}$. In all experiments, $T = 30$.

- **P2:** $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_j^b$ are interpolated for synthesizing new poses $\boldsymbol{y}_k^s$, where $k \in \{1, \cdots, N_I\}$. N_I is determined so that $\|\boldsymbol{y}_{k+1}^s - \boldsymbol{y}_k^s\| = y_{max}$ and $\ddot{\theta}_k^s = \ddot{\theta}_l^s$, where $\forall k, \forall l \in \{1, \cdots, N_I\}$. If $\ddot{\theta}_k^s$ is above $\ddot{\theta}_{max}^A$, $i = i - 1$ and $j = j + 1$ then start P2 again. Otherwise, the synthesized path is accepted. In the example in Fig. 7, the synthesized paths in (0) $i = j = 0$ and (1) $i = -1$ and $j = 1$ are rejected because $\ddot{\theta}_k^s$ is above $\ddot{\theta}_{max}^A$, while the path is accepted in (2) $i = -2$ and $j = 2$.

- **P3:** P2 synthesizes a path also from $\boldsymbol{y}_j^b$ to $\boldsymbol{y}_i^a$.

- **P4:** If two or more transition paths should be synthesized between around $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_j^b$, $i = i - 1$ and $j = j + 1$ and then go back to P2[3]

If the sample motions around the transition path are smooth, a smaller number of points are synthesized by the above mentioned interpolation. For example, since two dance actions have very similar poses (e.g. those indicated by green rectangles in Fig. 6), these poses were connected only with two or three synthesized poses, in experiments. Between walking and jogging actions, which were used also in the experiments, even the most similar poses are apart from each other, more poses were synthesized for each transition path; around five or more poses were synthesized.

The algorithm described above synthesizes pose data on a transition path. For accurate regression around the synthesized pose data, the regression function should be trained with these synthesized poses. For learning feature-to-pose regression with the synthesized poses, the feature data corresponding to them are required. The features corresponding to the synthesized

[3] In all experiments in this paper, only one path was synthesized between each pair of end points. The number of the paths should be determined in accordance with a task (e.g. applications, subjects, environments, and scenarios). For example, for human animations, motion transitions should be visually natural, but it is not understood how to quantify the amount of acceptable transitions, as described in [42]. In our proposed models for pose tracking, the number of the transition paths can be possibly determined automatically based on the distance between the end points; the shorter the distance is (i.e. the more similar the two actions a and b are), the more the number of the transition paths increase.

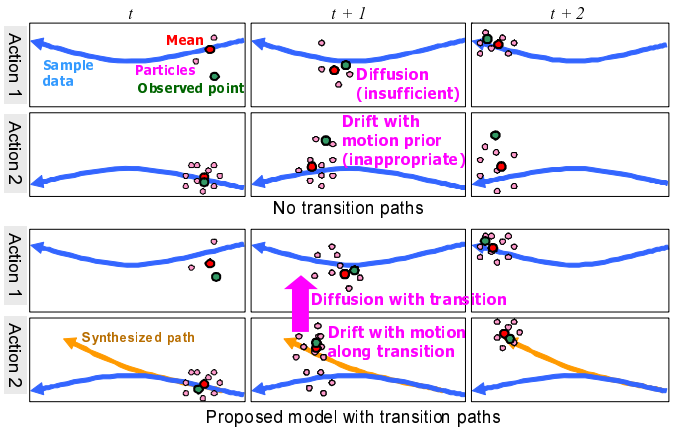


Figure 8: Particle transition between different actions. Upper and lower figures illustrate the histories of the particles with and without transition paths, respectively. Blue and yellow lines show latent variables of real sample data and synthesized data, respectively. Pink, red, and green points depict particles, their mean positions (i.e. current states), and data observed at each moment, respectively.

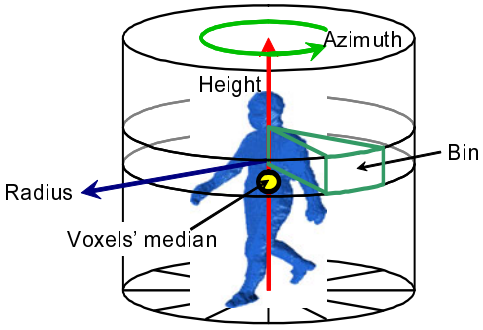


Figure 9: Bin structure in a volume descriptor.

poses are generated by interpolation of the features on the both ends under constraints given by the synthesized poses. The details of our implementation is described in Sec. 5.2.

4.3. Particle Tracking with Diffusion among Models along Synthesized Transition Paths

As described in Sec. 4.1, the basic scheme of our pose tracking is same with Condensation [40]. The main difference from Condensation is to distribute particles in multiple models and to propagate them among the models. Particle diffusion among the models is executed as follows:

1. In each model, all particles are diffused after drift using a motion prior. Diffusion is represented by the following equation:

$$p(\boldsymbol{p}^d|\boldsymbol{p}^p) \propto \exp\left(-\frac{1}{2}(\boldsymbol{p}^d - \boldsymbol{p}^p)^2\right), \quad (2)$$

   where $\boldsymbol{p}^p$ and $\boldsymbol{p}^d$ denote particles given by drift using the motion prior and the diffusion, respectively.
2. The particles each of whose nearest sample is one of the synthesized samples are selected. The nearest neighbor is found based on a Euclid distance in the latent space. Let m and n denote the model where the selected particle is located and the model to which m connects via the synthesized path, respectively.
3. For each of the selected particles, variance ${\sigma_m}^2$ in m is computed. The particle is mapped into n to compute ${\sigma_n}^2$ also. Then there is $\exp(-{\sigma_n}^2)$ in $(\exp(-{\sigma_m}^2) + \exp(-{\sigma_n}^2))$ chance that the particle moves to n.

A typical example of particle transition from actions 2 to 1 is illustrated in Fig. 8. If no transition path exists between actions 1 and 2, the number of particles in model 1 increases only when observed pose gets close to action 1 (i.e. at $t + 2$ in the figure). With the proposed model, on the other hand, the particles readily move to model 1 because of the synthesized path, which induces the particles to move to model 1 by diffusion. The synthesized path allows us also to obtain proper feature-to-pose regression during transition (i.e. at $t + 1$ in the figure) because the regression function is trained with the synthesized data along the path.

4.4. Scalable Motion Learning with Synthesized Paths

The learning scheme in our method achieves scalability against the number of actions by the following steps:

- **L1:** Given N_A action sequences. i is initialized to be 1.
- **L2:** For each of j-th actions s.t. $j < i$, pose and feature data on transition paths are synthesized between i-th and j-th actions. GPDM with shared latent modeling is applied to i-th sequence with all the synthesized data.
- **L3:** If $i < N_A$, $i = i + 1$ and then go back to L2. Otherwise, halt.

If a new $(N_A + 1)$-th action is added, perform L2 with $i = N_A + 1$. No re-optimization is required for other existing sequences. With this ad-hoc independent learning scheme, each model is optimized from only one action sequence and several synthesized data. Since the dominant computational cost in latent modeling is $O(N_S^3)$ [26] where N_S is the number of samples, independent learning reduces the computational cost significantly and thus achieves scalability of motion model learning.

5. Image Features and Their Interpolation

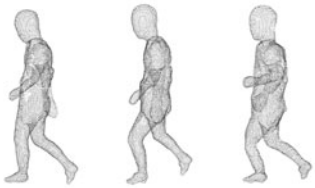
5.1. Image Features

Two kinds of image features were used for empirical evaluation in a studio. One of them was extracted only from a single view, and the other was from multiple views. The former was a set of shape contexts [43, 44], and the latter was a volume descriptor [48, 38], which is effective for reducing negative effects due to self occlusion in a single view.

Shape contexts [43] are log-polar histograms, each of whose center is a reference edge point, of the rest of edge points. The

walking sample synthesized pose jogging sample

(a) Edge points for computing shape contexts



walking sample synthesized pose jogging sample

(b) Surface points for computing a volume descriptor

Figure 10: Synthesizing image features along a transition path.

log-polar histogram has N^o orientation and N^r radius bins. In our experiments, the edge points were sampled from the boundary points of a subject's silhouette, which was extracted using background subtraction and color detection. The radius of the log-polar bins was normalized with respect to the size of the silhouette because the observed size of the subject varied depending on the distance from a camera.

The performance of shape contexts has been proved in comparative experiments [49]; since edge points could be extracted correctly in our experimental studio, shape contexts provided superior performance. Each shape was represented by a set of N^s shape contexts. In training, all shape contexts of all frames were divided to N^c codebooks by K-means clustering. Then shape contexts extracted at each frame were voted to the codebooks in order to make the histogram of the shape contexts. Each shape context is voted to the nearest neighbor codebook. This bag-of-features approach allows us to achieve robustness against local shape deformation and occlusion. In our experiments, N^o, N^r, N^s, and N^c were 12, 5, 200, and 100, respectively.

A volume descriptor [48, 38] is extracted from the visual hull of a subject, which is reconstructed by Shape-from-Silhouettes [45]. From the visual hull, surface voxels are extracted for efficient modeling, as with [48, 38]. The visual hull is then divided into several bins as shown in Fig. 9. In our experiments: 1) the shape of the bin model is a cylinder, whose vertical center line passes through the median of all human voxels; 16 height divisions and 36 azimuth divisions, and 2) the entities of each bin are the number of surface voxels and their mean distance from the center line. Unlike shape contexts, only one volume descriptor is extracted as a feature of each volume.

Both in shape contexts and volume descriptors, dissimilarity between two features, $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, is expressed by $\|\boldsymbol{f}_1 - \boldsymbol{f}_2\|^2$.

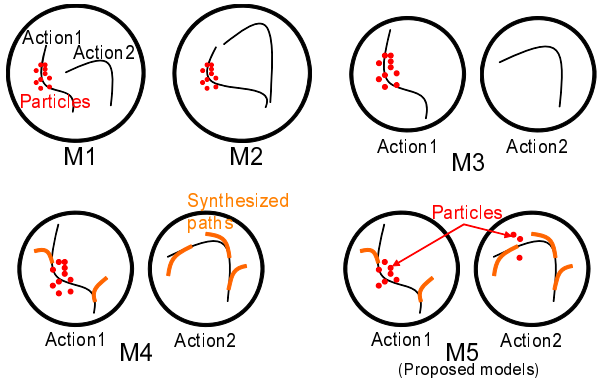


Figure 11: Five motion models used in experiments. Solid black and yellow lines depict real sample and synthesized data. Red dots depict particles. See the body text for the details of the five models.

5.2. Interpolating Features along Transition Paths

As described in Sec. 4.2.2, as well as pose data, image features must be synthesized along transitions paths. The features are synthesized by interpolation between two end points of the transition path.

Shape contexts corresponding to their respective synthesized pose are computed from edge points that are generated in accordance with the synthesized pose. For generating the edge points, the edge points extracted from the frames of two end points (denoted by end points a and b) are interpolated as follows. Given N^i synthesized frames between a and b and two sets of edge points extracted from a and b (denoted by edge sets E^a and E^b), point correspondence between E^a and E^b is first obtained. To this end, E^a and E^b are overlapped so that their centroids coincide with each other. Then, interpolation of $e^a \in E^a$ and $e^b \in E^b$, which are corresponding edge points, in i-th synthesized frame is computed as follows:

$$p(e,i) = \frac{\|J(i,b)\|}{\|J(i,a)\| + \|J(i,b)\|} D(i,a) + \frac{\|J(i,a)\|}{\|J(i,a)\| + \|J(i,b)\|} D(i,b), \quad (3)$$

$$J(i,q) = p(j(e^q), i) - p(j(e^q), q), \quad (4)$$

$$D(i,q) = p(e^q, q) + J(i,q), \quad (5)$$

$$q \in a, b$$

where $p(e,i)$ and $j(e)$ denote the position of e at i-th frame and the joint that is nearest to e, respectively.

In the same way as shape contexts, volume descriptors are synthesized by interpolating surface voxels instead of edge points.

Figure 10 shows examples of synthesized features between walking and jogging samples.

6. Experiments

6.1. Dataset for Evaluation

Synchronized video and pose datasets of multiple actions were used for learning and evaluation. Multiview videos were

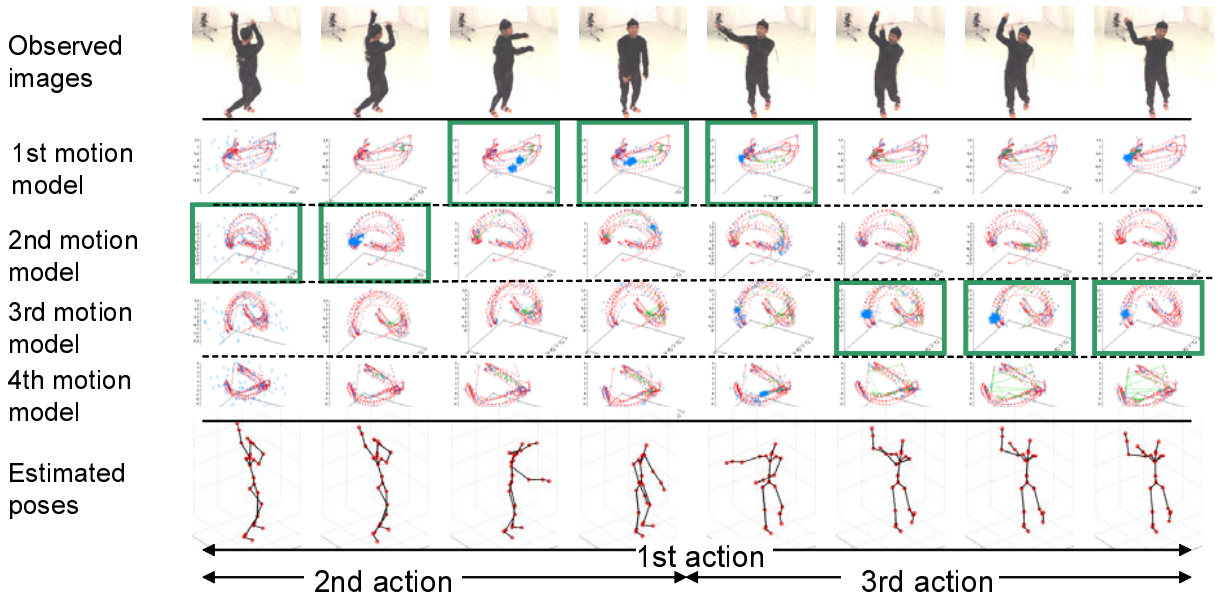


Figure 12: Pose tracking results with volume descriptors in action set1. Red and blue points in models (i.e. 2nd-5th rows) depict sample data and particles, respectively. The model selected for feature-to-pose regression at each moment is enclosed by a green rectangle.

captured by eight cameras at 30 fps (1024 × 768 pixels). For obtaining the ground-truth of the pose data, a gyro-sensor based motion capture system (IGS-190) was used. 51 dimensional pose data (i.e. 17 3-DOF joints) was obtained at each frame. Variables were set as follows throughout all experiments: w_v = 0.5, w_o = 0.5, the dimension of a latent space was 6, and the number of particles was 500.

While only one subject was captured for learning samples, four subjects were captured for testing data. With each subject, three kinds of action sets below were captured:

- **Set1 (four dance actions):** Waving the arms by different four ways: 1) "right-upper and left-upper", 2) "right-upper and left-lower", 3) "right-lower and left-upper", and 4) "right-lower and left-lower". All of them shared similar motion when the arms were waved in front of the body.
- **Set2 (two gait actions):** Walking and jogging actions.
- **Set3 (six gait actions):** 1) Walking, 2) walking slowly, 3) walking fast, 4) striding, 5) jogging, and 6) stopping from walking and start walking.

For evaluating the proposed models, these sequences are more suitable than existing video and mocap datasets in terms of including more actions in each sequence. For example, in HumanEva [4], only one transition from walking to jogging is appeared in each of six Combo sequences, which can be employed for training. Our dataset contains a number of transitions between each pair of actions, which are required for validating the effectiveness of the proposed model. 3–10 or more action transitions are contained in each sequence, which consists of more than 250 frames.

With each of the action sets, pose tracking was tested using two kinds of test sequences and five kinds of models:

- **Test sequence:** T1) only one of actions in each action set was performed through each sequence, where no action transition was observed, and T2) all actions and transitions between them were captured in each sequence. While T1 was used for evaluating the base accuracy of pose estimation, T2 was used for evaluating pose estimation accuracy during the action transitions and recovery from low accuracy after the transitions.
- **Motion models and particles (Fig. 11):** M1) all actions are modeled in a model with no synthesized paths, M2) topologically-constrained models proposed in [29], where different actions are modeled so that similar poses in the different actions are close to each other in the latent space[4], M3) actions are modeled in their respective models independently with no synthesized paths, M4) actions are modeled in their respective models independently with synthesized paths but using a unimodal motion prior at each moment, where all particles propagated in a single model at each moment, and M5) actions in their respective models with synthesized paths using the motion priors of multiple actions models (i.e. the proposed models).

The running times for unified modeling, M1, and independent modeling, M5, were 88 and 8 minutes, respectively, with T2 of set1. It is apparent that independent modeling (i.e. our model) is superior in scalability.

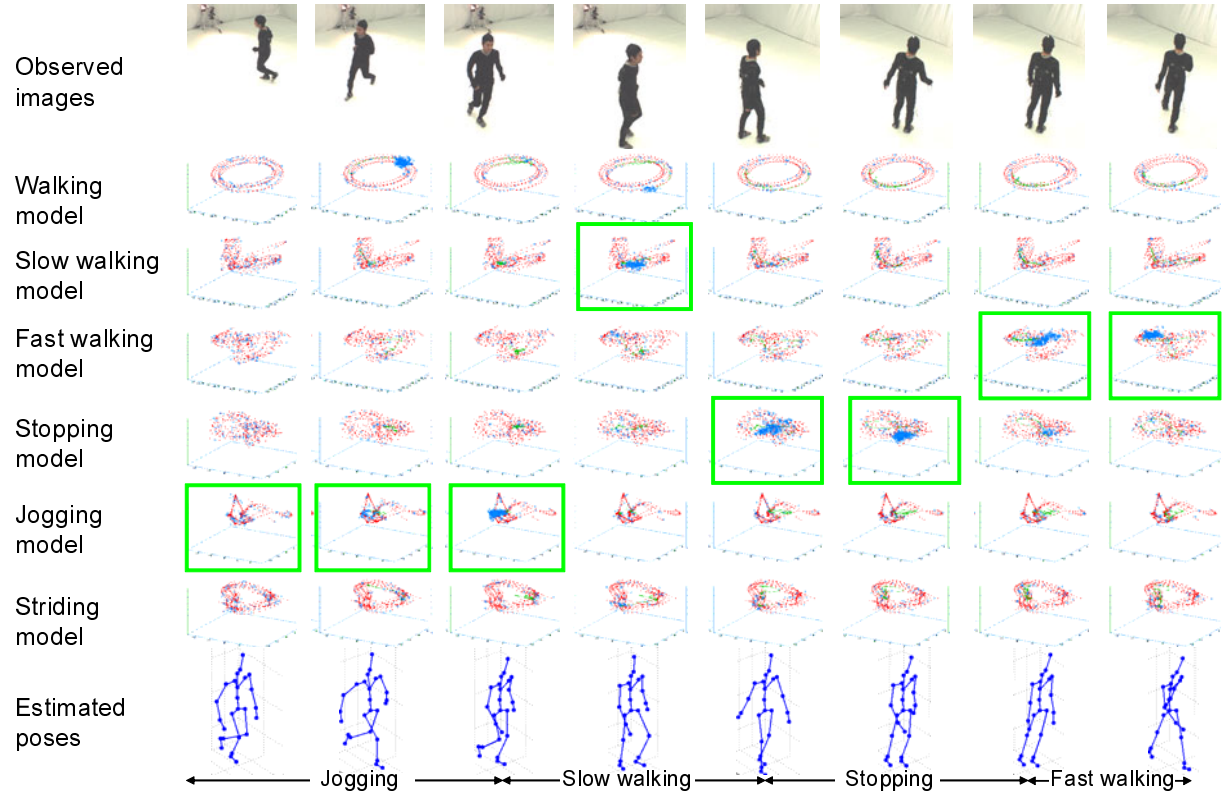


Figure 13: Pose tracking results with volume descriptors in action set3. Red and blue points in models (i.e. 2nd-7th rows) depict sample data and particles, respectively. The model selected for feature-to-pose regression at each moment is enclosed by a green rectangle.

6.2. *Pose Tracking with Volume Descriptors*

This section shows experimental results with volume descriptors, which are more discriminative than shape contexts.

Figures 12 and 13 show the tracking results with T2 of set1 and set3, respectively. In the figures, the latent space of each action model and particles in it are shown in the middle part. The leftmost images in each figure show initial frames, where particles were distributed almost uniformly in all motion models. The particles then gathered in a proper action model by following the motion of a subject. A correct action at each moment, which was given manually, is shown at the bottom in the figures. It can be seen that many particles were propagated in the model of the correct action observed at each moment. Note that, in the dance sequences (Fig. 12), the correct actions were overlapped because they shared similar motions.

The RMS errors of all joint positions through all frames are shown in table 1. The results of M3, M4, and M5, all of which are independent modeling but have different schemes for model transition, were similar to each other in T1 of all action sets. This is because no transition among actions was included in a test sequence of T1. In set1, even the errors of M1 and M2 (i.e. unified modeling) were not so worse than those of M3, M4, and M5. This is because only similar actions, which can be correctly optimized in a unified model, were observed. In set2, however, independent modeling (i.e. M4 and M5) was superior to unified modeling (i.e. M1 and M2) because of model optimization in each action. Note that independent modeling without transition paths (i.e. M3) was almost worse than M2. This might be because M3 has no mechanism for handling action transitions, while M2 enables smooth transition between different actions.

In set1, it can be also seen that the results of T2 & M4 and M5 (i.e. with transition paths) were almost same with that of T2 & M3 (i.e. with no transition paths). This might be happened because while T2 included action transitions, the different actions shared the very similar motions where model transitions

Table 1: RMS errors of estimated joints using volume descriptors.

(mm)	M1	M2	M3	M4	M5 (proposed)
Set1, T1	23	23	20	21	24
Set1, T2	30	25	28	26	26
Set2, T1	32	29	24	23	22
Set2, T2	39	38	34	30	25
Set3, T1	35	39	28	24	23
Set3, T2	41	34	37	31	23

[4] All similar poses of different actions were given manually.

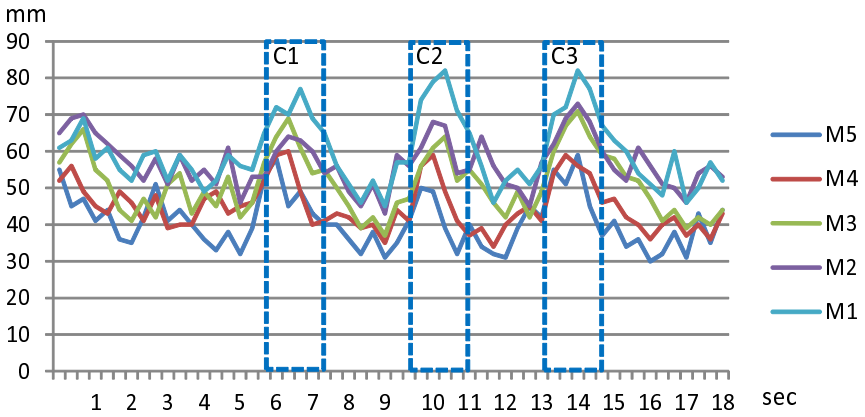


Figure 14: Comparison of pose tracking accuracy of the different five models. The graphs show the RMS errors of all joint positions estimated by using volume descriptors. Rectangles C1, C2, and C3 depict time intervals when action transitions are happened. For simplicity, the results at 3 fps are shown in the figure.

Table 2: RMS errors of estimated joints using shape contexts.

(mm)	M1	M2	M3	M4	M5 (proposed)
Set1, T1	53	52	44	40	40
Set1, T2	60	56	52	45	41
Set2, T1	48	50	46	44	41
Set2, T2	70	52	55	51	44
Set3, T1	72	76	63	59	47
Set3, T2	91	69	74	70	53

might happen; synthesized paths were not so useful in that case. On the other hand, in set2 and set3 where several action transitions cause between dissimilar poses, the errors in T2 & M5 were smaller than those in T2 & M3 and M4. This fact proved the effectiveness of synthesized paths and particle propagation in multiple models. In particular, the difference between the results of M3 and M5 proves the superiority of the proposed method in contrast to a method with model switching such as independent models proposed in [47].

Figure 14 shows temporal accuracy of pose estimation with the different five models. The graphs in the figure were obtained from T2–set3 of one subject. The results of this subject were worse than those of other subjects probably because his gait motion was relatively different from the training gait motion. Throughout this test sequences, it can be seen that all the models got worse results during action transitions (i.e. C1, C2, and C3 in the figure). It can be also seen that the models having the mechanisms for smooth action transitions (i.e. "M2 with topological constraints" and "M4 and M5 with synthesized transition paths") could recover from huge errors during the action transition, in contrast to other models, M1 and M3.

6.3. Pose Tracking with Shape Contexts

Pose tracking with shape contexts, which is more challenging than that with volume descriptors, was evaluated.

Figures 15 and 16 show tracking results of different subjects with set1 and set3, respectively. The RMS errors of all joint positions through all frames are shown in table 2. Examples of temporal pose estimation accuracy are also shown in Fig. 17. As with Fig. 14, the graphs in Fig. 17 were obtained from T2–set3 of one subject.

As expected, the results of all conditions were worse than those obtained by volume descriptors. Roughly speaking, inequality relations among the results obtained by shape contexts were almost similar to those obtained by volume descriptors; it can be seen that the results obtained by the proposed model and particle tracking (i.e. M5) were better than those of the others (i.e. M1, M2, M3, and M4).

7. Concluding Remarks

This paper proposed the motion models of multiple actions for human pose tracking. The models are acquired from independently captured action sequences so that potential transition paths between them are synthesized. Experimental results demonstrated that 1) independent modeling improves scalability of the modeling and 2) the synthesized paths and particles propagated in multiple models allow us to readily follow the change in action for correct pose tracking.

The proposed method synthesizes the transition paths based on simple criteria for pose similarity and basic interpolation between different poses. Additional constraints and criteria would improve the reasonability of the paths; for example, physical constraints for improving robustness and accuracy of detecting transition points [51] and pruning unrealistic motions [52].

For improving efficiency of particle propagation, additional constraints for propagating particles are effective; for example, human biomechanics [53]. More efficient algorithms for particle filtering would be also useful; for example, [9, 10, 46].

In the proposed method, transition paths are synthesized in the pose observation space. It is known that pose interpolation in a latent space is more accurate than that in its observation

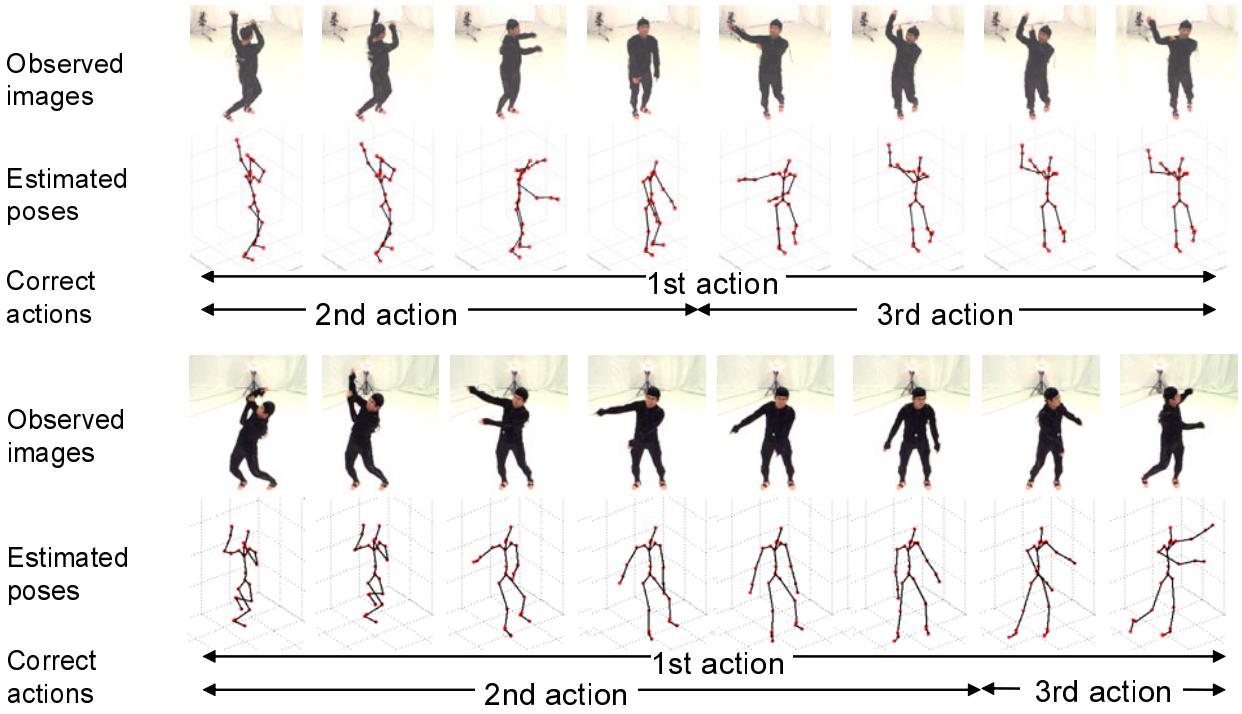


Figure 15: Pose tracking results of different subjects with shape contexts in action set1.

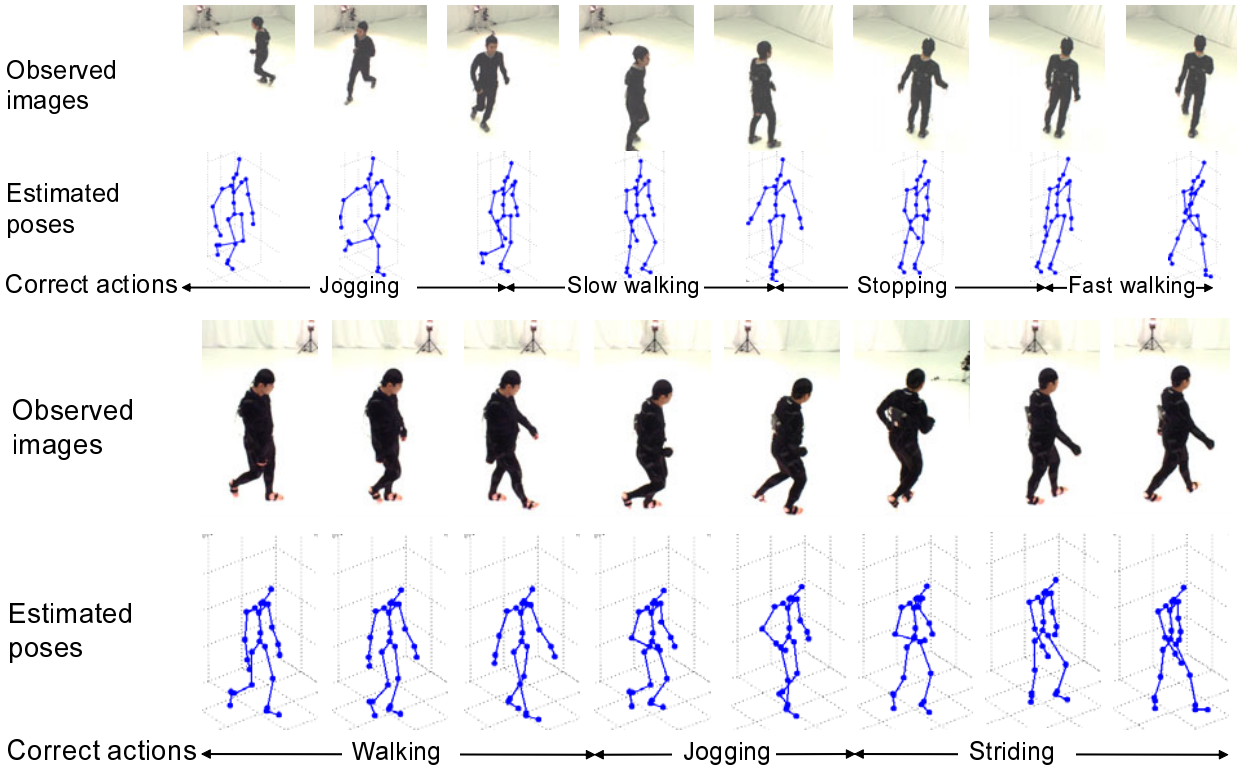


Figure 16: Pose tracking results different subjects with shape contexts in action set3.

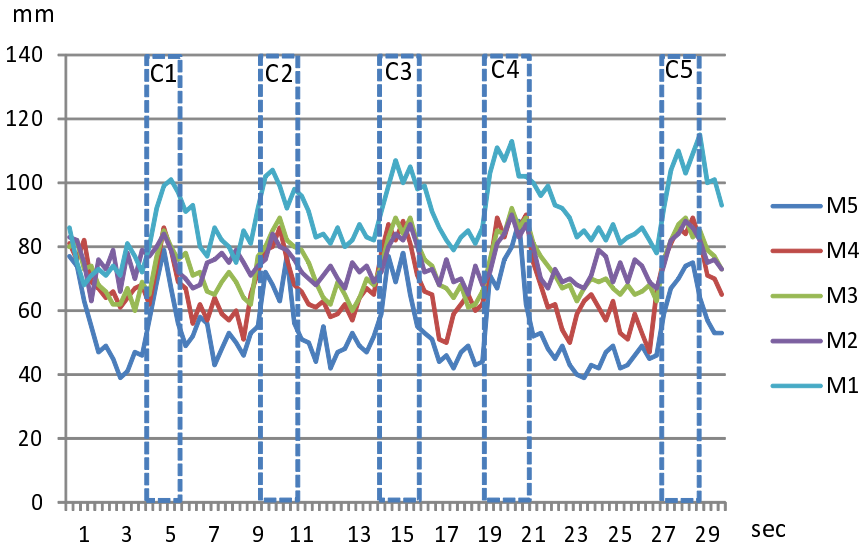


Figure 17: Comparison of pose tracking accuracy of the different five models. The graphs show the RMS errors of all joint positions estimated by using shape contexts. Rectangles C1, C2, C3, C4, and C5 depict time intervals when action transitions are happened.

space [50]. Increasing accuracy of path synthesis might improve pose tracking and regression during action transitions.

Furthermore, in our experiments, it was validated that similar actions (i.e. those in action set1) can be modeled even in a single model. The fewer the number of the models, the greater the number of particles in each model. This results in improving tracking stability. This fact suggests that the similar actions should be grouped and modeled together for combining the advantages of separate and unified modeling.

Acknowledgment

The GPDM codes were given by Neil Lawrence and Jack Wang. The authors would like to thank Takeo Kanade for his helpful comments.

References

[1] D. M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *CVIU*, Vol.73, No.1, pp.82–98, 1999.

[2] T. B. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *CVIU*, Vol.81, No.3, pp.231–268, 2001.

[3] R. Poppe, "Vision-based human motion analysis: An overview," *CVIU*, Vol.108, No.2, pp.4–18, 2007.

[4] L. Sigal and M. J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion," Techniacl Report CS-06-08, Brown University, 2006. http://vision.cs.brown.edu/humaneva/

[5] CMU Graphics Lab Motion Capture Database: http://mocap.cs.cmu.edu/

[6] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit Probabilistic Models of Human Motion for Synthesis and Tracking," *ECCV*, 2002.

[7] A. Agarwal and B. Triggs, "3D Human Pose from Silhouettes by Relevance Vector Regression," *CVPR*, 2004.

[8] R. Urtasun, D. Fleet, and P. Fua, "3D People Tracking with Gaussian Process Dynamical Models," *CVPR*, 2006.

[9] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects"' *ICCV*, 1999.

[10] J. Sullivan, A. Blake, M. Isard, and J. MacCormick, "Bayesian object localization in images," *IJCV*, Vo.44, pp.111–135, 2001.

[11] D. Vlasic, I. Baran, W. Matusik, and J. Popovic, "Articulated Mesh Animation from Multi-view Silhouettes," *ACM Transactions on Graphics*, Vol.27, No.3, 2008. http://people.csail.mit.edu/drdaniel/

[12] R. Urtasun, D. Fleet, and P. Fua, "Temporal motion models for monocular and multiview 3D human body tracking," *CVIU*, Vol.104, No.2, pp.157–177, 2006.

[13] N. How, M. Leventon, and W. Freeman, "Bayesian Reconstruction of 3D Human Motion from Single-Camera Video," *NIPS*, 1999.

[14] N. Huazhong, T. Tan, L. Wang, and W. Hu. "People tracking based on motion model and motion constraints with automatic initialization," *Pattern Recognition*, Vol.37, No.7, pp.1423–1440, 2004.

[15] C. Sminchisescu and A. Jepson, "Variational Mixture Smoothing for Non-Linear Dynamical Systems," *CVPR*, pp.608–615, 2004.

[16] M. Brand, "Shadow Puppetry," *ICCV*, 1999.

[17] S. Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley, "Real-time Body Tracking Using a Gaussian Process Latent Variable Model," *ICCV*, 2007.

[18] A. Elgammal and C.-S. Lee, "Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning," *CVPR*, 2004.

[19] A. Fathi and G. Mori, "Human Pose Estimation using Motion Exemplars," *ICCV*, 2007.

[20] X. Zhao and Y. Liu, "Tracking 3D Human Motion in Compact Base Space," *IEEE Workshop on Applications of Computer Vision*, 2007.

[21] T. Jaeggli, E. Koller-Meier, and L. "Multi-Activity Tracking in LLE Body Pose Space," *2nd Workshop on HUMAN MOTION Understanding, Modeling, Capture and Animation*, 2007.

[22] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, Vol.290, No.5500, pp.2319–2323, 2000.

[23] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, Vol.290, No.5500, pp.2323–2326, 2000.

[24] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput*, Vol. 26, No. 1, pp.313–338, 2004.

[25] S. Xiang, F. Nie, C. Pan, C. Zhang, "Regression Reformulations of LLE and LTSA with Locally Linear Transformation," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol.41, No.5, pp.1250–1262, 2011.

[26] N. D. Lawrence, "Probabilistic non-linear principal component analysis

with Gaussian process latent variable models," *Journal of Machine Learning Research*, Vol.6, pp.1783–1816, 2005.
[27] N. D. Lawrence, "Local distance preservation in the gp-lvm through back constraints," *ICML*, 2006.
[28] N. D. Lawrence and A. J. Moore, "Hierarchical Gaussian process latent variable models," *International Conference in Machine Learning*, 2007.
[29] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. Darrell, and N. D. Lawrence, "Topologically-Constrained Latent Variable Models," *ICML*, 2008.
[30] J. Chen, M. Kim, Y. Wang, and Q. Ji, "Switching Gaussian Process Dynamic Models for Simultaneous Composite Motion Tracking and Recognition," *CVPR*, 2009.
[31] J. M. Wang, D. J. Fleet, A. Hertzmann, "Multifactor Gaussian Process Models for Style-Content Separation," *ICML*, pp.975–982, 2007.
[32] N. Ukita and T. Kanade, "Gaussian Process Motion Graph Models for Smooth Transition among Multiple Actions," *CVIU*, Volume 116, Issue 4, pp.500-509, 2012.
[33] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes," ICML, 2009.
[34] L. Kovar, M. Gleicher, and F. H. Pighin, "Motion graphs," *SIGGRAPH*, 2002.
[35] J. M. Wang, D. J. Fleet, A. Hertzmann, "Gaussian Process Dynamical Models for Human Motion," *PAMI*, Vol.30, No.2, pp.283–298, 2008.
[36] C. H. Ek, P. H. S. Torr, and N. D. Lawrence, "Gaussian Process Latent Variable Models for Human Pose Estimation," *MLMI*, 2007.
[37] A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao, "Learning Shared Latent Structure for Image Synthesis and Robotic Imitation," *NIPS*, 2005.
[38] N. Ukita, M. Hirai, and M. Kidode, "Complex Volume and Pose Tracking with Probabilistic Dynamical Models and Visual Hull Constraints," *ICCV*, 2009.
[39] V. Pavlovic, J. M. Rehg, T.-J. Cham, and K. P. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," *ICCV*, 1999.
[40] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *IJCV*, Vol.29, No.1, pp.5–28, 1998.
[41] L. Zhao and A. Safonova, "Achieving good connectivity in motion graphs," *Graphical Models Journal*, Vol.71, No.4, pp.139–152, 2009.
[42] P. S. A. Reitsma and N. S. Pollard, "Evaluating Motion Graphs for Character Navigation," *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2004.
[43] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *PAMI*, Vol.24, No.4, pp.509–522, 2002.
[44] G. Mori, S. Belongie, and J. Malik, "Efficient Shape Matching Using Shape Contexts," *PAMI*, Vol.27, No.11, pp.1832–1837, 2005.
[45] G. Cheung, T. Kanade, J. Bouguet, and M. Holler, "A real time system for robust 3D voxel reconstruction of human motions," *CVPR*, 2000.
[46] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," *CVPR*, 2000.
[47] A. Geiger, R. Urtasun, and T. Darrell, "Rank Priors for Continuous Non-Linear Dimensionality Reduction,"*CVPR*, 2009.
[48] Y. Sun, M. Bray, A. Thayananthan, B. Yuanand, and P. H. S. Torr, "Regression-based human motion capture from voxel data," *BMVC*, 2006.
[49] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp.1615–1630, 2005.
[50] S. Bitzer and S. Vijayakumar, "Latent Spaces for Dynamic Movement Primitives," *IEEE RAS Humanoids*, 2009.
[51] A. Safonova, J. K. Hodgins, N. S. Pollard, "Synthesizing Physically Realistic Human Motion in Low-Dimensional, Behavior-Specific Spaces," *SIGGRAPH*, 2004.
[52] M. Vondrak, L. Sigal, and O. C. Jenkins, "Physical Simulation for Probabilistic Motion Tracking," *CVPR*, 2008.
[53] J. M. del Rincon, D. Makris, C. Orrite-Urunuela, and J.-C. Nebel, "Tracking Human Position and Lower Body Parts Using Kalman and Particle Filters Constrained by Human Biomechanics" *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 41, No. 1, pp.26–37, 2011.