Graphical Models 75 (2013) 328–345

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Reconstructing whole-body motions with wrist trajectories ☆

Hyejin Kim [a], Sung-Hee Lee [b,*]

[a] School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea

[b] Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

ARTICLE INFO

Article history:
Received 17 April 2013
Received in revised form 19 July 2013
Accepted 22 August 2013
Available online 11 September 2013

Keywords:
Character animation
Performance capture
Motion reconstruction
Low-dimensional control
Local linear model

ABSTRACT

Reconstructing whole-body motions using only a low-dimensional input reduces the cost of and efforts for performance capture significantly, and yet remains a challenging problem. We introduce a novel technique that synthesizes whole-body motion using the two wrist trajectories. Given the wrist trajectories, we first determine the optimal ankle trajectories from a large number of candidate ankle paths obtained from example poses in the motion database. The optimal trajectory is efficiently achieved by solving for the shortest path problem in a directed acyclic graph. Next, we use both the wrist and ankle trajectories as the low-dimensional control signals to achieve the whole-body pose at each time step. We show that our method can reconstruct various whole-body motions that can be recognized by arm motions, such as walking, stepping, and in-place upper-body motions. Comparisons with ground truth motions and with other methods are provided.

1. Introduction

Motion capture technologies are widely used in the computer gaming and movie industries because they allow for fast generation of realistic character animation as compared with the conventional key frame-based procedures. To make the motion capture-based animation processes more affordable and easier, researchers have put forth various efforts, ranging from the development of capturing devices [1–4] to editing techniques for captured signals [5–7].

To reduce the cost and clutter of attaching a number of sensors or markers to the human body, researchers have attempted to capture whole-body motion using only low-dimensional signals acquired from a sparse set of sensors and markers attached to end-effectors, such as hands and feet [8–10]. This approach aims to synthesize whole-body poses such that the non-captured parts are appropriately posed to realize the poses of the captured body parts. This approach is referred to as *performance (or action) capture* in contrast with the motion capture that aims to capture the exact configurations of every segment of a human body.

Obviously, the lower the dimension of captured signals, the harder it is to synthesize human motion. To tackle the problem, Chai and Hodgins [8] collect K-nearest poses from the pose database given the positions of six to nine markers attached to the end effectors and pelvis, and construct locally linear human pose space to determine a natural whole-body motion that satisfies the low-dimensional control signals. Liu et al. [11] synthesize whole-body motion with a dynamic motion model that estimates the current pose from the previously synthesized poses. Given the control sequences acquired from four accelerometers attached to the hands and feet, Tautges et al. [10] find pose candidates with higher temporal coherency using the novel Online Lazy Neighborhood Graph and synthesize whole-body motion from the noise-ridden accelerometer data.

Using a smaller number of sensors makes the problem even more challenging, especially if the sensory data for some end-effectors are unavailable. This paper presents a novel data-driven method to synthesize whole-body

☆ This paper has been recommended for acceptance by Nancy Pollard and Peter Lindstrom.

* Corresponding author.
E-mail addresses: hjinkim@gist.ac.kr (H. Kim), sunghee.lee@kaist.ac.kr (S.-H. Lee).

1524-0703/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.gmod.2013.08.002

animation from the trajectories of *only two wrist positions*. Because other body parts, such as the legs, can move more or less independently from the arms, it may seem an infeasible attempt to synthesize whole-body motion just from the two wrist positions, and apparently this approach cannot capture the whole range of human movement. However, a large portion of human motions (e.g., in-place upper-body motion, locomotion, jogging, jumping, etc.) can be reasonably estimated only from arm motions, and we seek to capture these motions. In fact, in many human activities, the hands have the utmost functional and semantic importance among all the human body parts, and the lower body only assists the hands to reach the target points. Therefore, our goal is to reconstruct natural whole-body motions that follow the input wrist trajectories, rather than accurately capturing whole-body movements.

Input to our method is the three-dimensional trajectories of two wrist points. These trajectories can be measured by externally installed stereo vision sensors, or by the cameras worn by the performer [4]. In our experiments, we acquire the input wrist trajectories by calculating the position of virtual markers attached to the wrist points in the test set of motion capture data and subsequently adding Gaussian noise to model the noise characteristics of real measurement data.

The advancement of tracking technologies has significantly improved the accuracy and computational load for estimating the cameras position and orientation [13,14]. Along with the wide dissemination of mobile devices equipped with vision and inertial sensors, our method can pave the way for enabling general public to capture their motions just by holding two general-purpose mobile devices in their hands, especially in outdoor environments where externally installed motion capture devices have limitations in capture volume. Additionally, our technique can be employed for existing motion capture devices to reconstruct whole-body motions when the lower-body is occluded by obstacles for a long duration.

Using the control signals of both the hands and feet, existing methods [8,10] show remarkable performances in generating natural-looking motions. However, these methods cannot be directly applied to the cases where only hand motions are used as the input. Because the correlation between hand and foot motions is rather low (especially if observed during a short time period), there are infinitely many foot configurations that can be associated with an input hand pose. Therefore, previous methods that determine optimal body motions only in terms of motion priors and smoothness may generate unnatural motions such as a foot skating on the ground or both feet floating in the air. This artifact should be addressed by taking into consideration the constraints on foot motions. Namely, the foot should not slide on the ground, and at least one foot must be in contact with the ground unless a jumping motion is exhibited. Additionally, the time interval for synthesizing motion should be long enough (at least longer than a single walking step) to achieve suitably correlated motions of the arms and the legs.

One approach to solving this problem is to add some naturalness constraints on the foot in the optimization process. However, solving such a spatial–temporal optimization problem to create a long duration of whole-body motions typically requires heavy computation and is not suitable for interactive applications.

Instead, we develop a novel two-step approach. First, using wrist trajectories acquired in a reasonably long duration, we determine only the associated optimal ankle trajectories by referring to motions in the database. Next, we use the acquired ankle trajectories, along with the wrist trajectories, as the low-dimensional control signals to achieve the optimal whole-body pose at each time step based on the technique proposed by Chai and Hodgins [8]. The advantages of this two-step approach are that (1) spatial–temporal optimization in a long time interval can be performed fast because the dimension of the unknown (i.e., the ankle trajectories) is low and (2) we can use the existing, highly-efficient methods [8,10] for synthesizing whole-body motion using the four control inputs. Fig. 1 shows the overview of our method. Given the captured wrist trajectories (Fig. 1(a)), the suitable ankle trajectories are estimated (Fig. 1(b)), and then the whole body motion is synthesized (Fig. 1(c) and (d)).

The key technical contribution of the proposed method is the procedure used to find the optimal ankle trajectories. We formulate the optimization process as a path-finding problem in a graph. At every time step $t(t = 1, \ldots, L)$, we collect K nearest poses from the database using the wrist points and identify the K candidate ankle positions corresponding to the poses. Next, we construct a directed acyclic graph (DAG) G in which each node $n_j^t(t = 1, \ldots, L, j = 1 \ldots K)$ represents the positions of the two ankles and edges (n_j^{t-1}, n_k^t) connect all nodes at time $t-1$ to the nodes at time t (Fig. 6). The cost of an edge $\left(n_j^{t-1}, n_k^t\right)$ is determined to represent the likeliness of the transition from ankle points n_j^{t-1} to n_k^t, and to penalize foot skating. Because of the structure of the DAG, the optimization is performed quite efficiently in time complexity $O(LK^2)$.[1] The ankle trajectories are further smoothed and used for the control input to find whole-body motion.

To complete the whole-body pose reconstruction, we follow the method of [8], with some key details added to reconstruct whole-body poses with only wrist information. Specifically, we do not only estimate joint angles, but also the optimal position and orientation of the pelvis, thereby allowing for natural pelvis movement without acquiring the sensory data about the pelvis. Previous methods either directly captured pelvis motion [8] or used the weighted average of pelvis poses from the motion database [10].

We show that our method can capture various performances, such as walking, sharp turns, or jumping as well as in-place upper-body motions. Comparisons are performed against the ground truth data and other related methods to verify the effectiveness of our approach.

The remainder of the paper proceeds as follows: after reviewing related work in Section 2, we present the overview of the proposed method in Section 3. Section 4 explains the preprocessing step, and Sections 5 and 6

[1] Finding a shortest path in a DAG is $O(V + E)$ where $V = LK$ is the number of vertices and $E = LK^2$ is the number of edges.

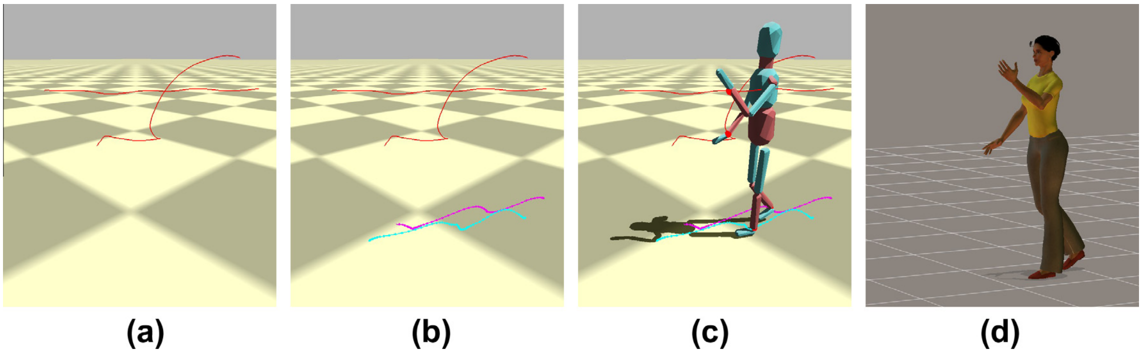


Fig. 1. Given the input trajectories of the wrist points (a), our method first estimates the ankle trajectories (b), and then synthesizes the whole-body motion satisfying both the wrist and ankle trajectories (c and d).

present the detailed procedures of our framework. After reporting the experiments in Section 7, we discuss the limitations and future work in Section 8. Section 9 concludes the paper.

2. Related work

Creating natural-looking animation of a character model with high degrees-of-freedom using low-dimensional input is a central problem in computer animation. Data-driven methods address this problem by using pre-recorded motion databases as motion prior [15–19]. A nave approach of searching the database for a sequence of best-matching motion clips and blending them is straightforward, but cannot satisfy the various constraints because of the limited size of the motion database. Instead, researchers have developed various techniques for creating new motions by interpolating and rearranging motion clips or by performing statistical modeling of poses or motions [20–23].

By predefining the existing natural transitions among the motion clips in the database using a graph, motion graph-based methods can create new, natural-looking permutations of motion clips. This approach efficiently finds the sub-optimal motion sequences to satisfy the imposed constraints, such as walking directions, by solving the optimization problem as a discrete path-finding problem [17,6,24]. However, motion graphs cannot create new poses that are not in the database; thus they cannot satisfy continuous, fine-grained constraints such as the hand and foot trajectories given in the performance capture.

In contrast, one can create a new motion that satisfies continuous constraints by interpolating motion clips in the database [5,25,26]. Additionally, by augmenting the motion graph with the motion interpolation technique [21], one can create a long sequence of motions that satisfy both the continuous (e.g., target hand trajectories) and discrete (e.g., target footprints) constraints. To this end, Zhao and Safonova [27] developed a method to increase good connectivity and transitions in the motion graph. However, non-trivial preprocessing for the motion clips, such as time warping, is required to interpolate the motions without creating visual artifacts.

Researchers have put steady efforts into building principled approaches to synthesizing natural human motions via statistical modeling of the human pose space. To model the high-dimensional, highly nonlinear human pose space, various methods have been applied, such as the mixture of Local Linear Models [28], Linear Dynamic System [29], and Gaussian Process-based models [30]. Chai and Hodgins [8] constructed locally linear human pose spaces to synthesize the natural human pose that works with the low-dimensional control inputs. Liu et al. [11] extended the approach to construct locally linear dynamic motion model online by finding a set of similar motion clips from the database and estimating current pose under the maximum a posteriori (MAP) framework. Krüger et al. [12] developed a novel lazy neighborhood graph for fast local and global similarity searching in a large motion capture database, and Tautges et al. [10] further improved the method to define the online lazy neighborhood graph that can find poses with higher temporal coherency from the database quite efficiently to construct locally linear pose spaces online. They showed that performance capture is somewhat possible even with an extremely low number (less than four) of sensors, but such performance is attainable only when the size of the database is small enough to ensure the discriminative ability of the sensor input.

Our method is mostly related to the work of Chai and Hodgins [8]. Our goal is to generate natural-looking whole-body motions using only the input signals from the two wrist trajectories. Because the dimension of input signal is very low, one cannot achieve the plausible motions by directly applying the methods in [8,10], which may create infeasible lower-body motions, such as foot skating, due to the lack of input signals involving the feet. To solve this problem, we construct feasible ankle trajectories for the input wrist trajectories and use both the wrist and ankle trajectories as the low-dimensional input to construct whole-body motions using the method of [8].

Recently the Gaussian process (GP) and the related methods are widely studied as a powerful means to reduce the high dimensional pose spaces to low-dimensional latent spaces [7,32,33,23,31]. The methods have several outstanding advantages, such as natural pose interpolation and less parameter-tuning. Grochow et al. [7] developed an inverse kinematics procedure that maintains the style of the example motions by using the GP latent variable model (GPLVM) [34]. Wang et al. [32] developed the GP dynamical models to model the temporal structure of motions. Levine et al. [31] proposed a method to generate interactive character animation that satisfies the user-specified inputs by precomputing the optimal control policies in the latent space of the GPLVM. GP-based methods are flexible enough to satisfy fine-grained, continuous constraints, yet they are limited to small-sized databases because of the cost of computation and training.

Among other interesting work that shares the goal of our method, Min et al. [35] developed a low-dimensional motion model tailored to support the geometric and timing variations of a motion and showed that the model can synthesize natural motions, satisfying both continuous and discrete constraints. In their work, separate motion models need to be constructed per each action category, which hinders continuous reconstruction of heterogeneous actions. The technique has been further improved to incorporate heterogeneous behaviors by concatenating distinctive motion models via graph walks and probabilistic sampling [36]. Slyper and Hodgins [2] developed a performance capture system that finds the best matching upper-body motion clip from the database using the input signals of five accelerometers sewn in a shirt. Yin and Pai [38] showed that a certain range of whole-body motion can be estimated just from the foot pressure distribution image. Heck et al. [37] presented a method to combine upper-body motion data from a motion sequence with the lower-body movement of another motion sequence in a natural manner. The latter two methods exploit the correlation between upper- and lower-body motions. We also use this feature but explore a different direction by estimating the feet movement from the input wrist motions.

3. Overview

As shown in Fig. 2, our performance capture framework consists of two major steps:

1. **Ankle trajectory generation**
   From the input trajectories of two wrist positions, we first estimate the trajectories for the ankles. Let $\boldsymbol{x}^t \in \mathbb{R}^6$ denote the thee-dimensional positions of the two wrist points, stacked to form a six-dimensional vector, at time t. The position of two ankle points at time t is similarly denoted as $\boldsymbol{y}^t \in \mathbb{R}^6$. Next, the input wrist trajectories are represented as $[\tilde{\boldsymbol{x}}^1, \ldots, \tilde{\boldsymbol{x}}^L]$, where the tilde sign "$\tilde{\cdot}$" denotes the input *control* signals.
   (a) **Making ankle point clouds**
   At each time step $t = 1, \ldots, L$, K_1 number of example poses, of which wrist positions are the closest to the given input $\tilde{\boldsymbol{x}}^t$, are selected from the database. Then, the ankle positions $\{\boldsymbol{y}_k^t | k = 1, \ldots, K_1\}$ corresponding to the selected poses form the *ankle point clouds* at each time step.
   (b) **Generating ankle trajectories**
   We construct a directed graph composed of all the ankle point clouds created throughout the time window $(1 \leqslant t \leqslant L)$, and determine the optimal ankle trajectories $[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L]$ by finding the shortest path in the graph.
2. **Whole body motion generation**
   At this step, we use both the wrist and ankle trajectories as the input control signals $[(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})^1, \ldots, (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})^L]$ to synthesize the whole-body pose at each time step based on the method of [8].
   (a) **Modeling a locally linear pose space**
   First, we construct the local linear model of the human pose using the K_2 closest poses $\{\mathbf{m}_k | k = 1, \ldots, K_2\}$ corresponding to the control signal $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})^t$.
   (b) **Generating whole-body poses**
   The local linear model is used to synthesize the natural-looking whole-body pose (i.e., the position and orientation of the pelvis as well as the joint angles) that aligns with the control signal.

The generated motion is finally post-processed to remove possible foot-skating artifact and improve smoothness.

4. Preprocessing the motion database

Given a motion database, we pre-compute several features for each pose in the database to perform the capture efficiently. The features are measured with respect to the *reference frame* defined for each pose based on the position of the wrists (Fig. 3). The reference frame is defined such that the x and z-coordinates are in the middle of the two

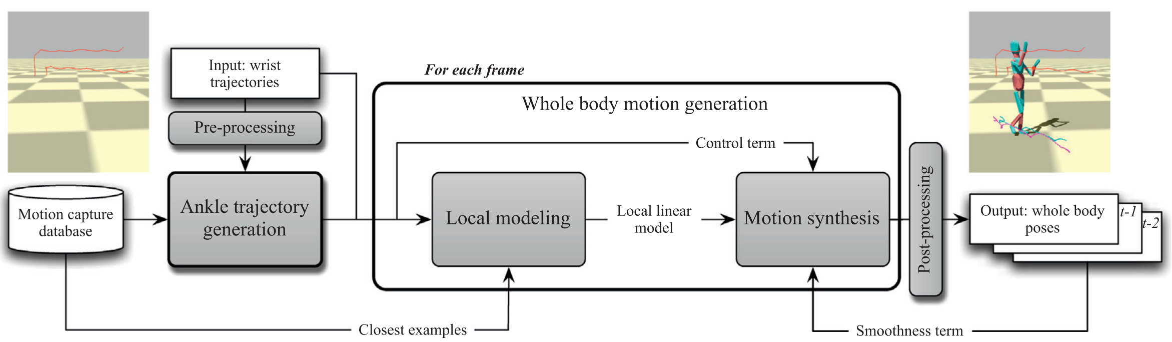


Fig. 2. Overview of the performance capture framework.

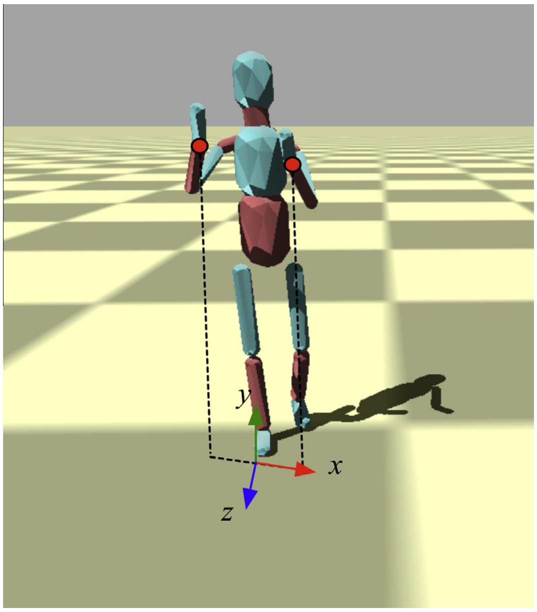


Fig. 3. The reference frame of a pose is determined from the wrist positions projected to the ground.

wrist points and the y-coordinate is at ground height. The orientation of the reference frame is aligned such that its x-axis is parallel to the line segment made by the two ground-projected points of the wrists, and the y-axis is same as the up-vector. The motivation for choosing the particular reference frame is that we want to find the K-nearest poses from the database with respect to the wrist positions, as will be explained in the next section.

Specifically, we pre-compute the following features: the axial tilt angle (angle between the y-axis of the reference frame and that of the pelvis) θ and the height h of the pelvis and the position of the wrists and ankles $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, as well as their velocities $(\dot{\hat{\boldsymbol{x}}}, \dot{\hat{\boldsymbol{y}}})$; all are expressed with respect to the reference frame. The hat sign "ˆ" denotes that the quantity is expressed in the reference frame. The velocities of the wrists and ankles are calculated by the numerical differentiation between the consecutive poses with respect to the world frame, and subsequently transformed to the reference frame. The height and tilt angle of the pelvis will be used to model the local linear pose space, along with the joint angles. The positions and velocities of the wrists and ankles are used to find close poses to the input signals.

5. Ankle trajectory generation

The challenge of reconstructing a whole-body pose from only wrist points is due to the fact that there is a wide variety of poses that would satisfy the given wrist points at a certain time step; thus, it is quite difficult to find temporally coherent and natural-looking motions if the poses are generated on a per-frame basis. However, if the wrist trajectories are examined within a sufficiently long time interval, a reasonable whole-body motion that matches the input wrist trajectories can be found. For example, if the wrist points move continuously within a certain range in space, they can be recognized as an in-place upper-body motion. If the wrist points translate over a long distance, they can be interpreted as locomotion.

Therefore, we first aim to reconstruct the plausible ankle trajectories by examining the wrist trajectories during a sufficiently long time interval. Then both the wrist and ankle trajectories will be used to reconstruct the whole-body motion in the next step.

The generation of the ankle trajectories is illustrated in Fig. 4. At a certain instant, one can find a subset of poses from a motion database that have similar wrist positions to the given wrist positions, and subsequently collect the set of candidate ankle points for the left and right feet from the poses. Typically the candidate ankle points are located close to each other, forming a cloud (hence, dubbed *ankle point cloud* in this paper). As these ankle point clouds are found at every point in time, the temporal evolution of the shape of the point clouds is acquired. We assume that feasible ankle trajectories can be found from the moving point clouds, and determine the optimal ankle points at time t from the point clouds at that particular time step.

We formulate the problem of finding the optimal ankle trajectories into a path finding problem in a graph. The node of the graph represents a pair of candidate ankle points. If we consider the time interval of $t = 1, \ldots, L$ and find K pairs of candidate ankle points at each time frame, a total of LK nodes are obtained. From our assumption, a node at time $t-1$ can only be connected to the nodes at t, hence the graph is constructed as a DAG. We find the optimal ankle trajectories by finding the shortest path in a DAG, which can be performed very efficiently.

5.1. Preprocessing

Because the motion data measured by capture devices contain noises, we first use a smoothing filter to reduce jittering of the measured data. In our experiment, we employ a simple exponential smoothing [41] to calculate the position $\tilde{\boldsymbol{x}}^t$ of wrist points at time t:

$$\tilde{\boldsymbol{x}}^t = s\underline{\boldsymbol{x}}^t + (1-s)\tilde{\boldsymbol{x}}^{t-1}, \tag{1}$$

where $\underline{\boldsymbol{x}}^t$ denotes the wrist positions at time t measured by capture devices, and s is a smoothing weight. We set $s = 0.5$ in our experiments.

5.2. Ankle point clouds

At each time step, ankle point clouds are created by the K_1 closest poses $p_k(k = 1, \ldots, K_1)$ found from the motion database using a pair of wrist points. Specifically, given the wrist position and velocity $(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}})^t$ at time t, we use the following metric to find the K_1 closest poses:

$$\alpha\|\tilde{\boldsymbol{x}}^t - \boldsymbol{x}_n\|^2 + (1-\alpha)\|\dot{\tilde{\boldsymbol{x}}}^t - \dot{\boldsymbol{x}}_n\|^2 \tag{2}$$

where $\boldsymbol{x}_n$ and $\dot{\boldsymbol{x}}_n$ denote the wrist positions and velocities of the pose $p_n(n = 1, \ldots, N)$, where N is the total number of poses in the database, aligned to the input wrist points (Fig. 5(a)). The alignment is performed by transforming the reference frame of p_n to coincide with a reference

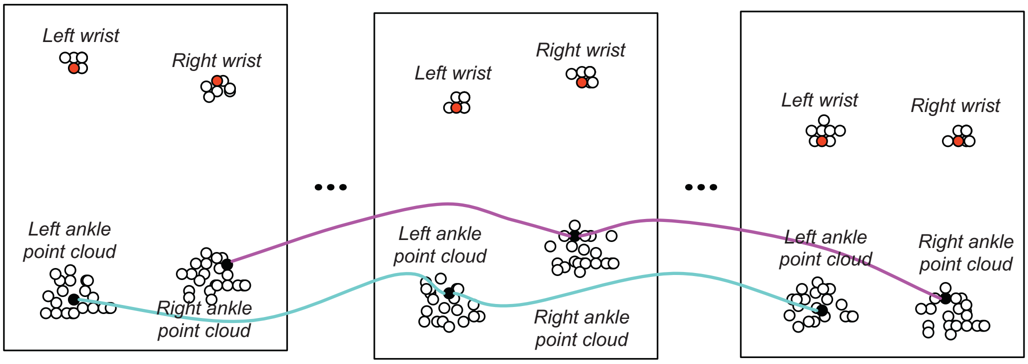


Fig. 4. The optimal ankle trajectory is created from the ankle point clouds, with the ankle points at time t selected from the point clouds at time t.

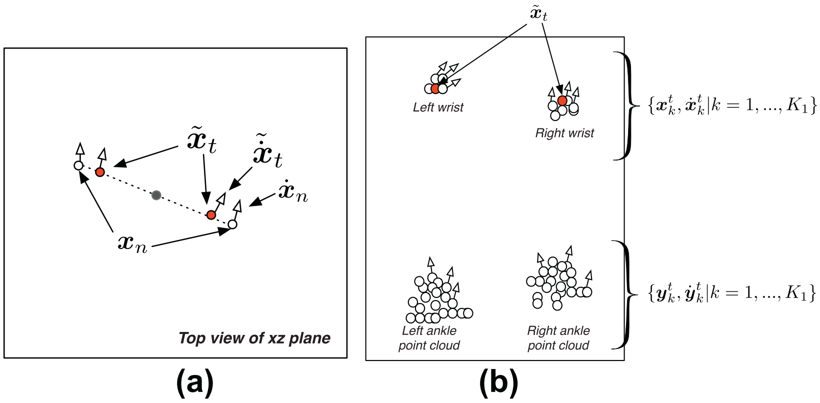


Fig. 5. Alignment of input and database wrist positions (left), and generation of ankle point clouds (right).

frame defined by $\tilde{\boldsymbol{x}}^t$. $\dot{\tilde{\boldsymbol{x}}}^t$ is calculated as the numerical differentiation between adjacent points. In practice, rather than aligning all the example poses to the given control input, we obtain an equivalent result for Eq. (2) more quickly by calculating the local coordinates of the input wrist points and velocities with respect to their reference frame and comparing them with $\hat{\boldsymbol{x}}_n$ and $\dot{\hat{\boldsymbol{x}}}_n$.

The first term in Eq. (2) checks how the position of the input control signals match the aligned position of the example pose in the database, and the second term compares the similarity of the velocities. The latter term differentiates between poses with similar wrist positions but different wrist velocities and helps to distinguish between the different behaviors when finding the example ankle points. For example, considering the static standing pose, the velocities of the input wrists are low, and in this case, the ankle points with high velocities, which were captured from walking or running motions, are less likely to be selected because their corresponding wrist motions will have high velocities.

After collecting K_1 number of poses that are closest to the input wrist points in terms of Eq. (2), we construct the ankle point clouds $\{\boldsymbol{y}_k^t | k = 1, \ldots, K_1\}$ at time $t = 1, \ldots, L$ (Fig. 5(b)). In our experiments, α and K_1 are set to 0.8 and 1000, respectively.

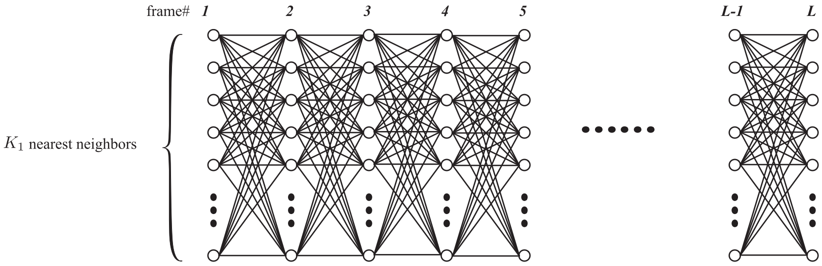
5.3. Optimal ankle trajectories

Using the ankle point clouds at time $t = 1, \ldots, L$, we find the optimal ankle trajectories from the example points in the clouds. To this end, we first construct a DAG, of which the nodes are the ankle point clouds $\{\boldsymbol{y}_k^t | t = 1, \ldots, L, k = 1, \ldots, K_1\}$. Edges are created such that all nodes at time $t-1$ are connected to all nodes at time t, i.e., $(\boldsymbol{y}_i^{t-1}, \boldsymbol{y}_j^t) \in E, \forall i, j, t$, where E denotes the set of edges in the graph (Fig. 6(a)).

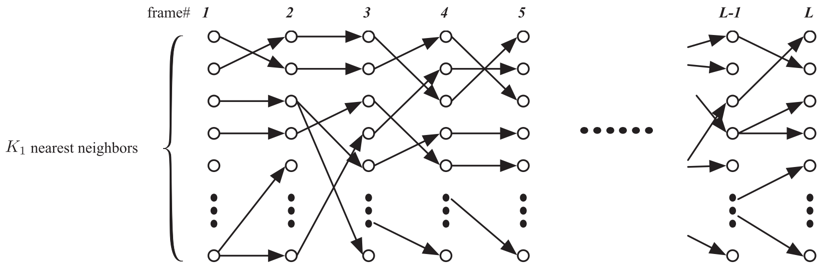
To determine the optimal ankle trajectory by finding the shortest path in the graph, we need to define suitable costs for each edge in the graph. For this, we consider two terms: *motion prior* and *sliding* terms. By forming the ankle point clouds, we have already collected the likely positions of the ankles using the input wrist points. What we need to consider next is how likely a path (i.e., an ankle trajectory) in a DAG is, which is what motion prior term deals with. Specifically, we define the cost $\sigma_{j,k}^{t-1}$ of an edge $\left(\boldsymbol{y}_j^{t-1}, \boldsymbol{y}_k^t\right)$ for the motion prior term as follows:

$$\sigma_{j,k}^{t-1} = \|\boldsymbol{y}_j^{t-1} + \Delta t \dot{\boldsymbol{y}}_j^{t-1} - \boldsymbol{y}_k^t\|^2, \tag{3}$$

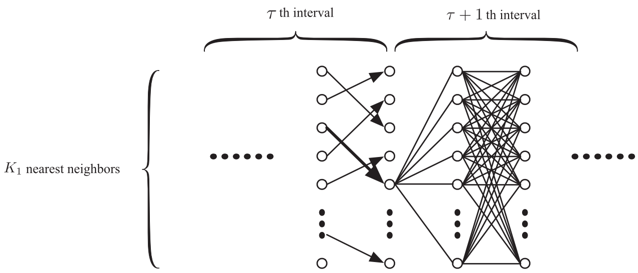
where Δt is the duration of a time step and $\boldsymbol{y}_j^{t-1} + \Delta t \dot{\boldsymbol{y}}_j^{t-1}$ represents the position that $\boldsymbol{y}_j^{t-1}$ would likely to evolve at



(a) A directed acyclic graph built from the ankle point clouds from frames 1 to L.



(b) Result of running Algorithm 1. We choose the path of the minimum cost.



(c) The continuity between the shortest paths in consecutive intervals is achieved by removing all outgoing edges from the nodes at the initial boundary of the interval $\tau+1$ except for the node included in the optimal path (thick arrow) in the τ-th interval.

Fig. 6. The optimal ankle trajectories are found by searching the shortest path in a directed acyclic graph.

the next time step. Hence, $\sigma_{j,k}^{t-1}$ represents the difference between the likely position of $\boldsymbol{y}_j^{t-1}$ at time t and $\boldsymbol{y}_k^t$. The smaller $\sigma_{j,k}^{t-1}$ is, the more likely that $\boldsymbol{y}_j^{t-1}$ will evolve to $\boldsymbol{y}_k^t$.

Another important criterion is whether the ankle position is physically plausible. The sliding term penalizes foot-sliding movement using the following cost:

$$d_{j,k}^{t-1} = d_{j,k}^{l,t-1} + d_{j,k}^{r,t-1}, \tag{4}$$

$$d_{j,k}^{l,t-1} = \begin{cases} \exp\left(w\|\boldsymbol{y}_j^{l,t-1} - \boldsymbol{y}_k^{l,t}\|^2\right), & \text{if } \mathit{sliding}\left(\boldsymbol{y}_j^{l,t-1}, \boldsymbol{y}_k^{l,t}\right) = \text{true} \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $\boldsymbol{y}_j^l$ denotes the left ankle part in $\boldsymbol{y}_j$. The cost grows exponentially as the distance between the ankle points increases. We determine $\mathit{sliding}\left(\boldsymbol{y}_j^l, \boldsymbol{y}_k^l\right)$ to be true if (1) the distance between $\boldsymbol{y}_j^l$ and $\boldsymbol{y}_k^l$ exceeds 2 cm and (2) the height difference of $\boldsymbol{y}_j^l$ and $\boldsymbol{y}_k^l$ is less than 0.14 cm. $d_{j,k}^{r,t-1}$ is defined similarly to $d_{j,k}^{l,t-1}$. Note that we do not check the distance between the ankles and the ground to detect the sliding because we also want to prevent the artifact that the feet slide in the air.

Overall, the cost $w_{j,k}^{t-1}$ of an edge $\left(\boldsymbol{y}_j^{t-1}, \boldsymbol{y}_k^t\right)$ is determined as the weighted sum of the two terms:

$$w_{j,k}^{t-1} = \beta\sigma_{j,k}^{t-1} + (1-\beta)d_{j,k}^{t-1}, \tag{6}$$

where β controls the weight between $\sigma_{j,k}^{t-1}$ and $d_{j,k}^{t-1}$. We set $\beta = 0.5$ in our experiments.

Finally, by finding the shortest path in the DAG, we determine the optimal ankle trajectory $[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L]$ that minimizes the following objective function:

$$[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L] = \arg\min_{\{\boldsymbol{y}^t, 1 \leqslant t \leqslant L\}} \sum_{i=1}^{L} w_{j,k}^{t-1}. \tag{7}$$

Algorithm 1 presents the pseudocode to find the optimal ankle trajectories in the graph. Fig. 6(b) shows the result after all paths in the DAG have been evaluated. The path of the minimum cost is chosen and the optimal nodes are collected by back-tracing the path.

Algorithm 1. GenerateAnkleTrajectory $(L, \{\boldsymbol{y}_j^t | t = 1, \ldots, L, j = 1, \ldots, K_1\})$

Input: Trajectory length L, the set of aligned ankle positions in motion capture database for L frames $\{\boldsymbol{y}_j^t | t = 1, \ldots, L, j = 1, \ldots, K_1\}$

Output: Two ankle positions as estimated control signals $[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L]$

1: **for all** $t = 1, \ldots, L-1$ **do**
2:   **for all** $j = 0, \ldots, K_1 - 1$ **do**
3:     **for all** $k = 0, \ldots, K_1 - 1$ **do**
4:       $\sigma_{j,k}^{t-1} = \|\boldsymbol{y}_j^{t-1} + \Delta t \dot{\boldsymbol{y}}_j^{t-1} - \boldsymbol{y}_k^t\|^2$
5:       $d_{j,k}^{t-1} = 0$
6:       **if** $\mathit{sliding}\left(\boldsymbol{y}_j^{l,t-1}, \boldsymbol{y}_k^{l,t}\right) = \mathit{true}$ **then**
7:         $d_{j,k}^{t-1} = d_{j,k}^{t-1} + \exp\left(w\|\boldsymbol{y}_j^{l,t-1} - \boldsymbol{y}_k^{l,t}\|^2\right)$
8:       **if** $\mathit{sliding}\left(\boldsymbol{y}_j^{r,t-1}, \boldsymbol{y}_k^{r,t}\right) = \mathit{true}$ **then**
9:         $d_{j,k}^{t-1} = d_{j,k}^{t-1} + \exp\left(w\|\boldsymbol{y}_j^{r,t-1} - \boldsymbol{y}_k^{r,t}\|^2\right)$
10:       $w_{j,k} = \beta\sigma_{j,k}^{t-1} + (1-\beta)d_{j,k}^{t-1}$
11:       **if** $n_k^t.\mathit{estimate} > n_j^{t-1}.\mathit{estimate} + w_{j,k}$ **then**
12:         $n_k^t.\mathit{estimate} = n_j^{t-1}.\mathit{estimate} + w_{j,k}$
13:         $n_k^t.\mathit{predecessor} = j$
14: $[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L] = \mathit{BackTraceDirectAcyclicGraph}()$
15: **return** $[\tilde{\boldsymbol{y}}^1, \ldots, \tilde{\boldsymbol{y}}^L]$

The acquired trajectory is smoothed by the Gaussian filter and used as the control signal for reconstructing whole-body motion in the next step.

5.3.1. Optimization in sub-intervals

Let us assume a real-time application that reconstructs the performer's motion in an interactive rate. As the wrist positions are input to the system in real-time, we can add K_1 number of nodes to the DAG and update the costs of the paths in the graph online (from line 1 to 12 in Algorithm 1). However, since the optimal path can only be identified after the costs of all the paths are finalized by back tracing the path (line 13 in Algorithm 1), one should wait until the end of performance before she can acquire the optimal ankle trajectories.

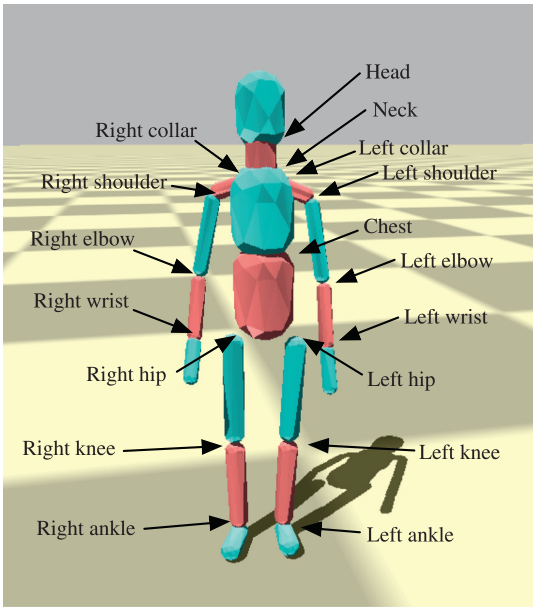


Fig. 7. A character model used in our experiments.

Instead, in order to achieve results in an interactive rate, we can choose to partition the graph into a set of intervals, and find the optimal trajectories in each interval of the graph. In other words, as a certain time interval L of wrist points are gathered, we perform the optimization in that interval, and wait for the next input wrist trajectories of length L to construct the subsequent optimal ankle path. This per-interval basis reconstruction creates sub-optimal results, but our experiments show that the sub-optimal results are not significantly different from the globally optimal path if the time interval L is long enough (Fig. 12). This is because the optimal foot motion for the given wrist positions at a certain time step can be identified in the sufficiently long, yet local time interval.

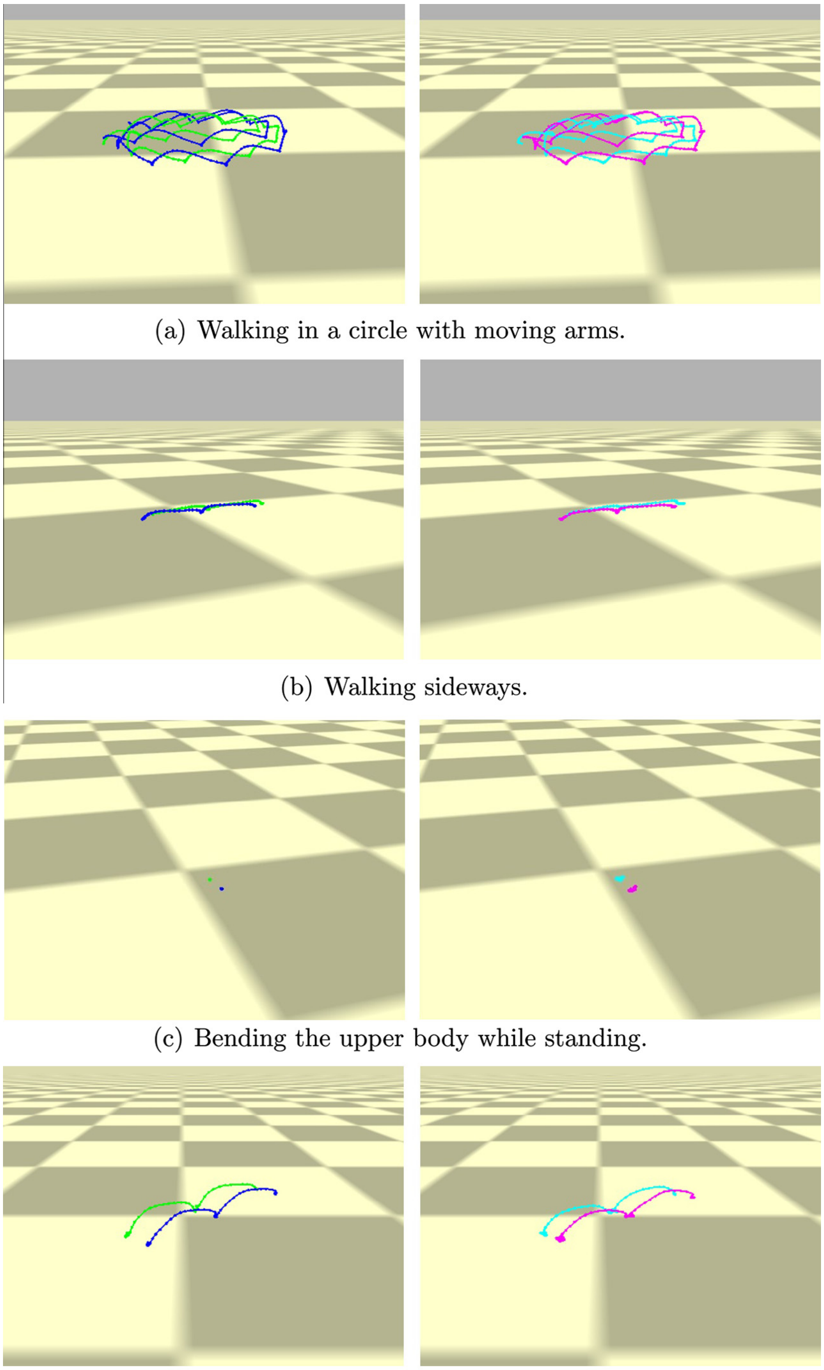
To ensure the continuity of the optimal paths between consecutive intervals, we enforce the last node of the optimal path in the previous time interval τ to be the first node of the optimal path in the next interval $\tau + 1$, by removing all the outgoing edges from the nodes at the initial boundary of the interval $\tau + 1$ except for the optimal node (Fig. 6(c)).

6. Motion generation

We reconstruct a whole-body pose for each time step; first, by constructing the locally linear pose space and the reduced pose vector, and second, by solving for the optimal pose given the control input.

6.1. Local modeling of poses

We construct the local linear model with the example poses collected with the control inputs. The overall process is based on [8], yet some key modifications and



(a) Walking in a circle with moving arms.

(b) Walking sideways.

(c) Bending the upper body while standing.

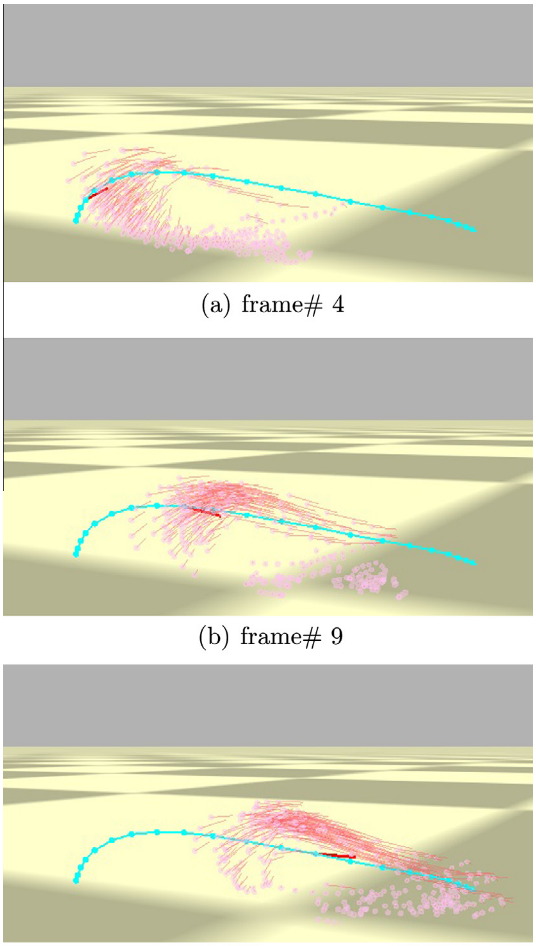
(d) Jumping twice.

Fig. 8. Comparisons of ground truth ankle trajectories (left) with the estimated ones (right). Green and cyan curves represent the left ankle trajectories, and blue and magenta curves represent the right ankle trajectories. See Fig. 14 for input wrist trajectories for the above results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

improvements are made to include the pelvis pose in the linear model.

Because of the ankle trajectories acquired in the previous step, we have four control signals with which we can find pose examples from the database to learn the pose space. A total of K_2 example poses are collected from the database with the following query metric:

$$\alpha\left(\|\tilde{\boldsymbol{x}}^t - \boldsymbol{x}_n\|^2 + \|\tilde{\boldsymbol{y}}^t - \boldsymbol{y}_n\|^2\right) + (1-\alpha)\left(\|\dot{\tilde{\boldsymbol{x}}}^t - \dot{\boldsymbol{x}}_n\|^2 + \|\dot{\tilde{\boldsymbol{y}}}^t - \dot{\boldsymbol{y}}_n\|^2\right). \quad (8)$$



(a) frame# 4

(b) frame# 9

(c) frame# 13

Fig. 9. The shapes of the left ankle point clouds for a walking motion at three different time frames. Only 500 points are visualized for clarity. The red line segments represent the velocities of each point, and the thick red line segment is the velocity of the node selected in the optimal path. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Note that $\boldsymbol{x}_n, \dot{\boldsymbol{x}}_n, \boldsymbol{y}_n$, and $\dot{\boldsymbol{y}}_n$ are the quantities corresponding to the pose aligned to the input wrist points in the same manner shown in Fig. 5(a). Eq. (8) is similar to Eq. (2), with the additional terms for the ankle inputs. α is set to the same value as in Eq. (2).

Let us denote a pose with $p = (\boldsymbol{z}, \mathbf{q})$, where $\boldsymbol{z} = (\boldsymbol{z}_R, \boldsymbol{z}_P) \in SE(3)$ is the position $\boldsymbol{z}_P$ and orientation $\boldsymbol{z}_R$ of the pelvis, and $\mathbf{q} \in \mathbb{R}^D$ is the joint angles of D degrees of freedom ($D = 54$ in our experiment). To construct the pose space, we define the feature vector for a pose as $\mathbf{m} = (\theta, h, \mathbf{q}^T)^T \in \mathbb{R}^{2+D}$, where θ and h are the axial tilt angle (the angle between the up-vectors of the world and the pelvis) and height of the pelvis, respectively. Note that, among the six degrees of freedom of the pelvis, we keep only the height and the tilt angle because they are invariant to the specific location and direction of the pose data in the database.

To find the reduced linear space, principal component analysis is used for the K_2 number of example poses $\{\mathbf{m}_n^t | n = 1, \ldots, K_2\}$ and a pose is represented using the low-dimensional reduced vector $\mathbf{w}^t$:

$$\mathbf{m} = \mathbf{p}^t + \boldsymbol{U}^t \mathbf{w}^t, \tag{9}$$

where $\mathbf{p}^t$ is the mean vector of the K_2 poses, $\boldsymbol{U}^t$ consists of D' number of principal components, and $\mathbf{w}^t \in \mathbb{R}^{D'}$ is the reduced space vector for the pose. In our experiment, we used the values of $K_2 = 200$ and $D' = 6$.

6.2. Motion synthesis

After constructing the linear pose space, we determine the whole-body pose $p^t = (\boldsymbol{z}, \mathbf{q})^t$ at each time step $(1 \leqslant t \leqslant L)$ by achieving the optimal values for the reduced vector $\mathbf{w}^t$ and the pose of the pelvis $\boldsymbol{z}^t$. This is achieved by solving an optimization problem to minimize the cost terms regarding the naturalness of the motion and satisfaction of the control inputs. Each cost term is provided in detail.

- **The constraint terms**
  The terms enforce constraints due to the structure and dependencies of the variables. First, $\boldsymbol{z}_R$ must satisfy the SO(3) structure. If represented with a unit quaternion, $\boldsymbol{z}_R$ must satisfy $(\boldsymbol{z}_R)^T \boldsymbol{z}_R = 1$. Also, the tilt angle θ and height h of the pelvis, defined by the reduced pose vector $\mathbf{w}$, should be in accordance with $\boldsymbol{z}$, that is, $\theta = \theta|_{\boldsymbol{z}}$ and $h = h|_{\boldsymbol{z}}$, where $\theta|_{\boldsymbol{z}}$ and $h|_{\boldsymbol{z}}$ are the values computed from $\boldsymbol{z}$.
  Combining the constraints above, we define the following constraint terms:

$$E_{const} = \| (\boldsymbol{z}_R^t)^T \boldsymbol{z}_R^t - 1 \|^2 + w_\theta \| \theta^t - \theta|_{\boldsymbol{z}^t} \|^2 + w_h \| h^t - h|_{\boldsymbol{z}^t} \|^2. \tag{10}$$

We used the values $w_\theta = w_h = 1$ in our experiments.

- **The prior term**
  Following [8], we enforce that the poses conform to the normal distribution made by the K_2 example poses, by applying the prior term defined as:

$$E_{prior} = (\mathbf{m}^t - \mathbf{p}^t)^T (\boldsymbol{\Lambda}^t)^{-1} (\mathbf{m}^t - \mathbf{p}^t), \tag{11}$$

where $\boldsymbol{\Lambda}^t$ is the covariance matrix of the K_2 example poses (see [8] for more details).

- **The control term**
  The control term deals with the quality of satisfying the control inputs:

$$E_{ctrl} = \| \boldsymbol{x}|_{\mathbf{w}^t, \boldsymbol{z}^t} - \tilde{\boldsymbol{x}}^t \|^2 + \| \boldsymbol{y}|_{\mathbf{w}^t, \boldsymbol{z}^t} - \tilde{\boldsymbol{y}}^t \|^2, \tag{12}$$

where $\boldsymbol{x}|_{\mathbf{w},\boldsymbol{z}}$ and $\boldsymbol{y}|_{\mathbf{w},\boldsymbol{z}}$ denote $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, given $\mathbf{w}$ and $\boldsymbol{z}$, computed by a forward kinematics operation.

- **The smoothness term**
  This term plays the role of increasing the smoothness of the resulting motion. Assuming that movement with constant velocity is the ideal case, i.e., $\mathbf{q}^t - \mathbf{q}^{t-1} = \mathbf{q}^{t-1} - \mathbf{q}^{t-2}$ for the joints and similarly for the pelvis, the smoothness terms is defined as:

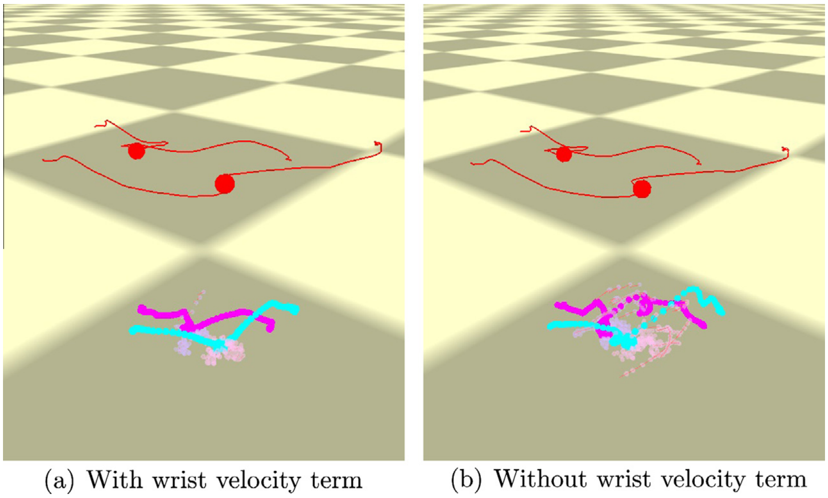


Fig. 10. The resulting ankle trajectories and the shape of point clouds at some moment when velocity metric in Eq. (2) is used (left) and when not (right). Cyan and magenta curves represent left and right ankle trajectories, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

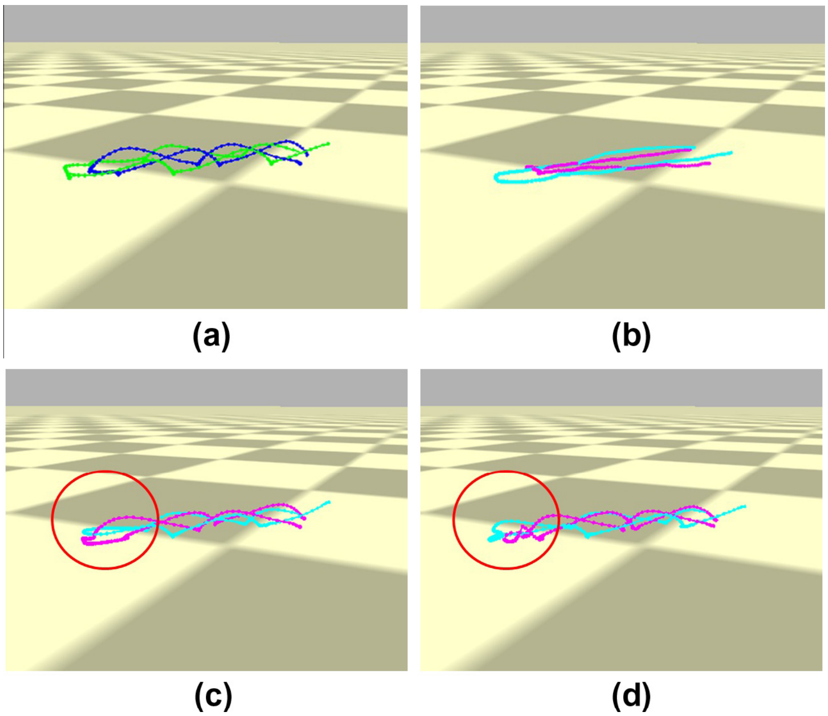


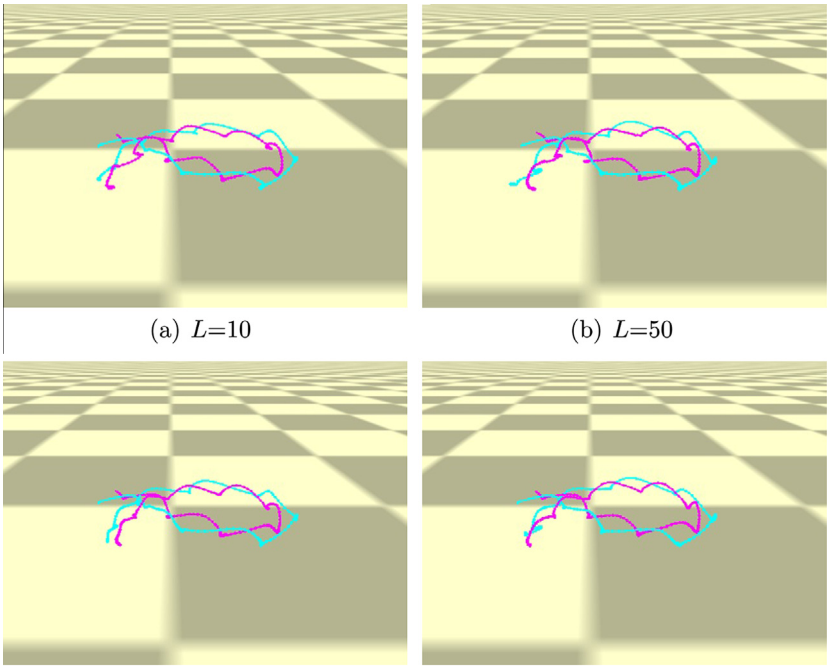
Fig. 11. The resulting ankle trajectories with different edge costs in the directed acyclic graph. The ground truth trajectory (walking to the left-sharp turn-walking to the right) is shown in (a). Green and blue curves represent the left and right ankle trajectories, respectively. We test cost functions of (b) the Euclidean distance between the nodes (c) prior term only (d) both the prior and sliding terms. The foot skating observed in (c) (in a red circle) is removed in (d). Cyan and magenta curves are the left and right ankle trajectories, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$E_{smooth} = \|\boldsymbol{z}^t - 2\boldsymbol{z}^{t-1} + \boldsymbol{z}^{t-2}\|^2 + \|\mathbf{q}^t - 2\mathbf{q}^{t-1} + \mathbf{q}^{t-2}\|^2. \tag{13}$$

The objective function is then defined as the weighted sum of all terms:

$$\arg\min_{\mathbf{w}_t, \boldsymbol{z}_t} E_{const} + w_{prior}E_{prior} + w_{ctrl}E_{ctrl} + w_{smooth}E_{smooth}. \tag{14}$$

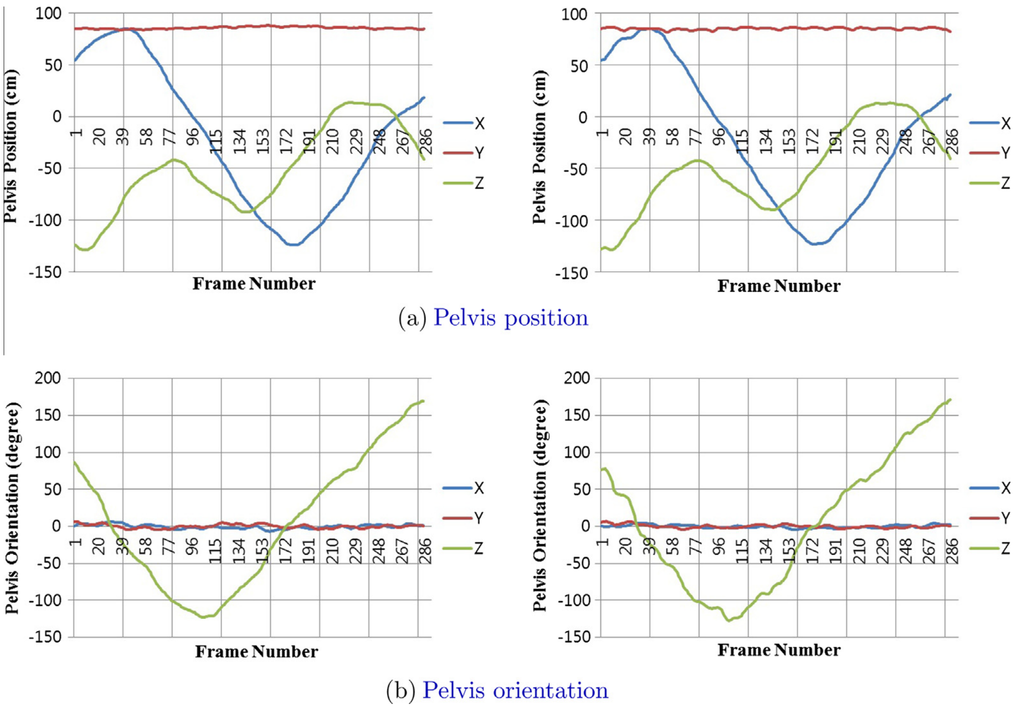
Eq. (14) is solved using the Levenberg–Marquardt algorithm [40]. The initial values for the optimization parameters $\mathbf{w}^t$ and $\boldsymbol{z}^t$ are set to the optimal values at $t-1$ for $t \geqslant 2$. At $t = 1$, the initial values are determined by the initially guessed pose, which is selected from the database to be the closest to $\boldsymbol{x}^t$, and then properly aligned. In our experiments, we set $w_{prior} = 1.0$, $w_{ctrl} = 0.8$, and $w_{smooth} = 0.2$.



(a) L=10 (b) L=50

(c) L=100 (d) L=total frames

Fig. 12. The resulting ankle trajectories with different lengths of time intervals L for a circular walking motion. Cyan and magenta curves are the left and right ankle trajectories, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Pelvis position

(b) Pelvis orientation

Fig. 13. Comparison of the ground truth (left) and the estimated pelvis motion (right) for a walking motion.

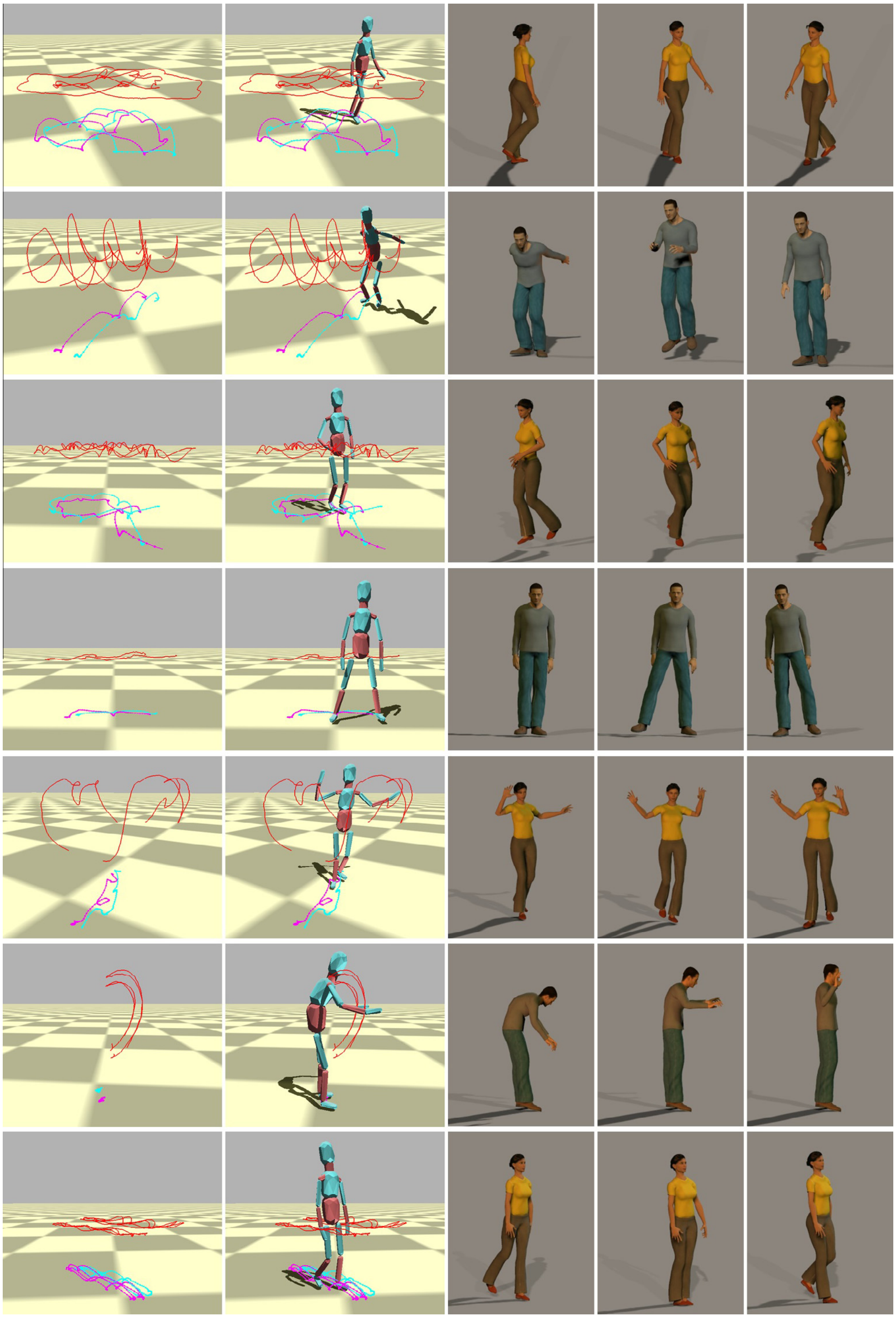


Fig. 14. Examples of the reconstructed motions. From the top, normal walking with swinging arms (#1), jumping twice (#2), jogging (#3), lateral stepping (#4), walking with waving hands (#5), bending the upper body while standing (#6), and forward stepping followed by backward stepping (#7).

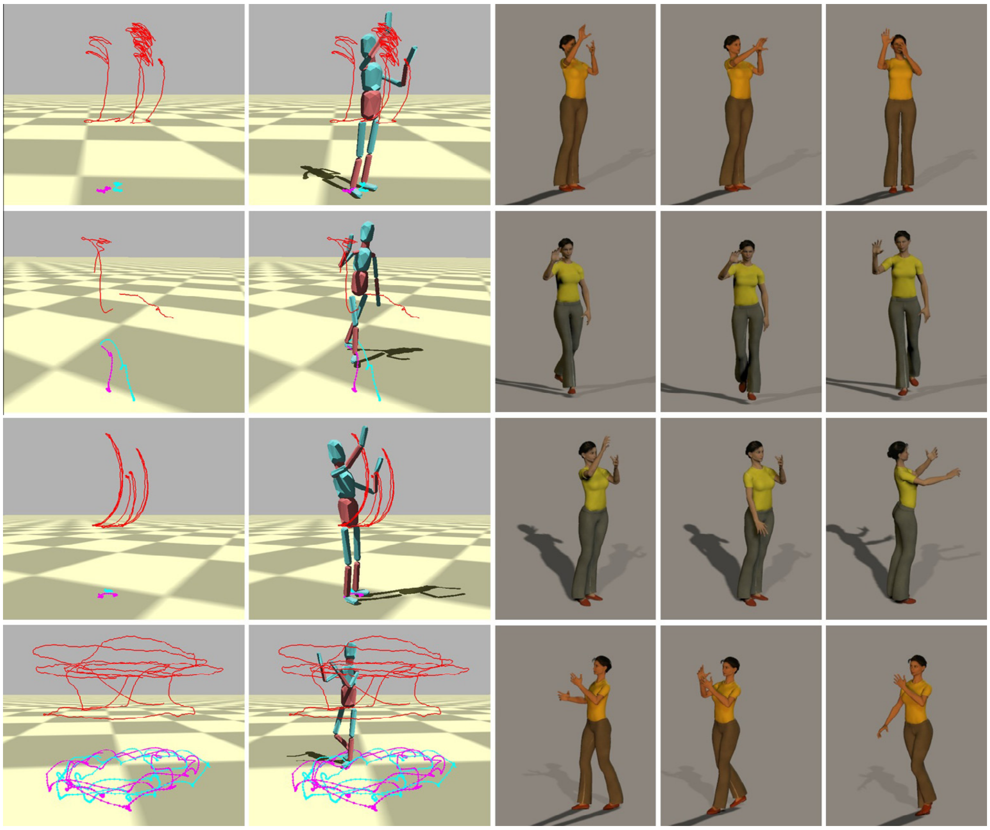


Fig. 15. Different lower-body movements with similar arm motions are well distinguished in our method. Top two rows: greeting motions (#8 and #9). Bottom two rows: hands are waving vertically (#10 and #11).

6.3. Post-processing

In most cases, the obtained whole-body motion looks natural and follows the input wrist trajectories well. However, in some cases, the motion contains slightly jittery movements when the ankle trajectories are not smooth enough. In addition, the resulting motion may contain foot-skating especially during a challenging motion such as a sharp-turn.

Therefore, we perform the post-processing to reduce the jittering and foot-skating artifacts. We first reduce the foot-skating using the technique presented in [39]. After identifying the desired configuration of a foot in contact with the ground, inverse kinematics is performed to constrain the foot to the desired configuration. Next, we apply the Gaussian filter in the joint space to improve the smoothness of the motion.

7. Experiments

We performed various experiments to verify the validity and effectiveness of our method. The motion database consists of a set of full body behaviors (a total of 49,709 frames running 28 min with 30 frames per seconds) such as standing with upper body motions (15,137 frames), normal walking with swinging arms (9350 frames), walking with other upper body motions (15,028 frames), running (2655 frames), jumping (2011 frames), walking sideways (2635 frames), bending the upper body (857 frames), and backward stepping (2036 frames).

Test input trajectories of the wrist points are obtained from separately captured motions doing the same behaviors in the motion database. We attach virtual markers to the wrist points for the test set of motions and acquire wrist trajectories using forward kinematics. Subsequently, we add Gaussian noise with standard deviation of 0.5 cm to the wrist trajectories to include the effect of the noises in the experiments. The human model used in our experiments has 17 joints, all of which are modeled as ball joints (Fig. 7).

7.1. Ankle trajectory generation

Fig. 8 shows the ankle trajectories achieved by our method for various input wrist trajectories including normal walking (with swinging arms), sideways walking, forward bending, and jumping. The figure shows that our method creates reasonably similar patterns to the ground truth ankle trajectories.

Fig. 9 shows the shape of the ankle point clouds and the velocities of each point (red line segments) at three different time frames in a walking motion. The shape of the point clouds varies significantly with each behavior and for each phase of the wrist in the same behavior. Note that

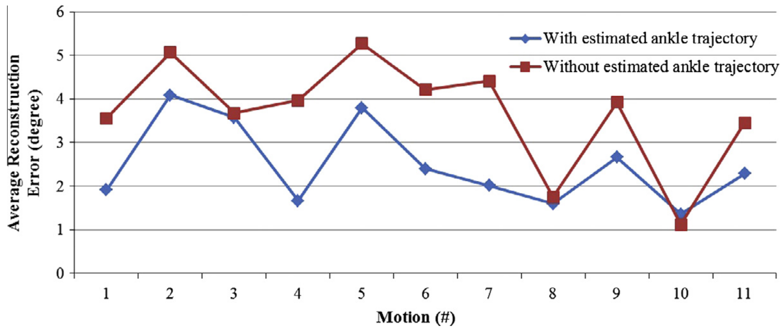


Fig. 16. Average reconstruction errors in each example.

the velocity vector $\tilde{\boldsymbol{y}}^t$, marked as a thick red line segment, of a node in the optimal ankle path (cyan) may not be collinear to the line segment $\overrightarrow{\tilde{\boldsymbol{y}}^t\tilde{\boldsymbol{y}}^{t+1}}$ made between itself and the next optimal node. This implies that the next optimal node $\tilde{\boldsymbol{y}}^{t+1}$ belongs to a pose that is not consecutive to that of the current node in the database.

The velocity metric in Eq. (2) helps to find candidate ankle points with higher temporal correlation. Fig. 10 shows the comparison between cases where the velocity metric is used and those where it is not used. When the velocity metric is used, the point cloud is more concentrated and the resulting ankle trajectory is smoother.

Fig. 11 evaluates the different cost functions for the edge in finding the optimal ankle trajectories in the graph. If the distance between the nodes $\left(\|\boldsymbol{y}_j^{t-1}-\boldsymbol{y}_k^t\|^2\right)$ is used for the cost function $w_{j,k}^{i-1}$, the shortest path in the sense of Euclidean distance is achieved rather than a natural human walking trajectory (Fig. 11(b)). If only the motion prior term is used without the sliding term, foot skating is not prevented during a sharp turn (Fig. 11(c) and (d) shows the resulting ankle trajectory. Although not exactly same as the ground truth data (Fig. 11(a)), a plausible ankle path is achieved.

We investigated the effect of length of the time interval on the optimization of the ankle trajectories. If the time interval is too narrow (Fig. 12(a)), the quality of the resulting trajectory is rather poor because the correlation between the ankle and the wrist is not evident over a short time period. If the time interval is longer than 50 frames, the results for the different window sizes are quite similar.

7.2. Pelvis estimation

In order to verify how accurately the pelvis position and orientation are estimated, we compare them with the ground truth values in the case of normal walking motion, as shown in Fig. 13. The reconstructed pelvis motion has a similar pattern as that of the ground truth, with a slight jitter noticed in the orientation.

7.3. Whole body motion reconstruction

Fig. 14 shows the snapshots of the reconstructed motions for various input wrist trajectories. Walking (#1), jumping (#2), running (#3), walking sideways (#4), walking with hand waving (#5), upper-body bending (#6), and forward stepping followed by backward stepping (#7) motions are successfully reconstructed. The proposed algorithm distinguishes between different lower-body motions of similar arm movement patterns well due to the velocity metric in Eq. (2). Fig. 15 shows such examples, for greeting motions while standing (#8) and walking (#9), and vertical hand-waving while standing (#10) and walking (#11).

Fig. 16 shows the accuracy of the reconstruction in terms of the average root-mean-square (RMS) error per

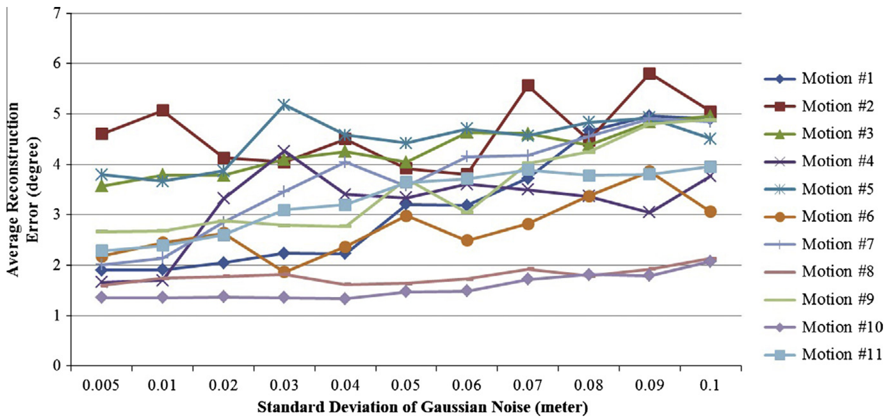


Fig. 17. Effect of noise levels on reconstruction errors.

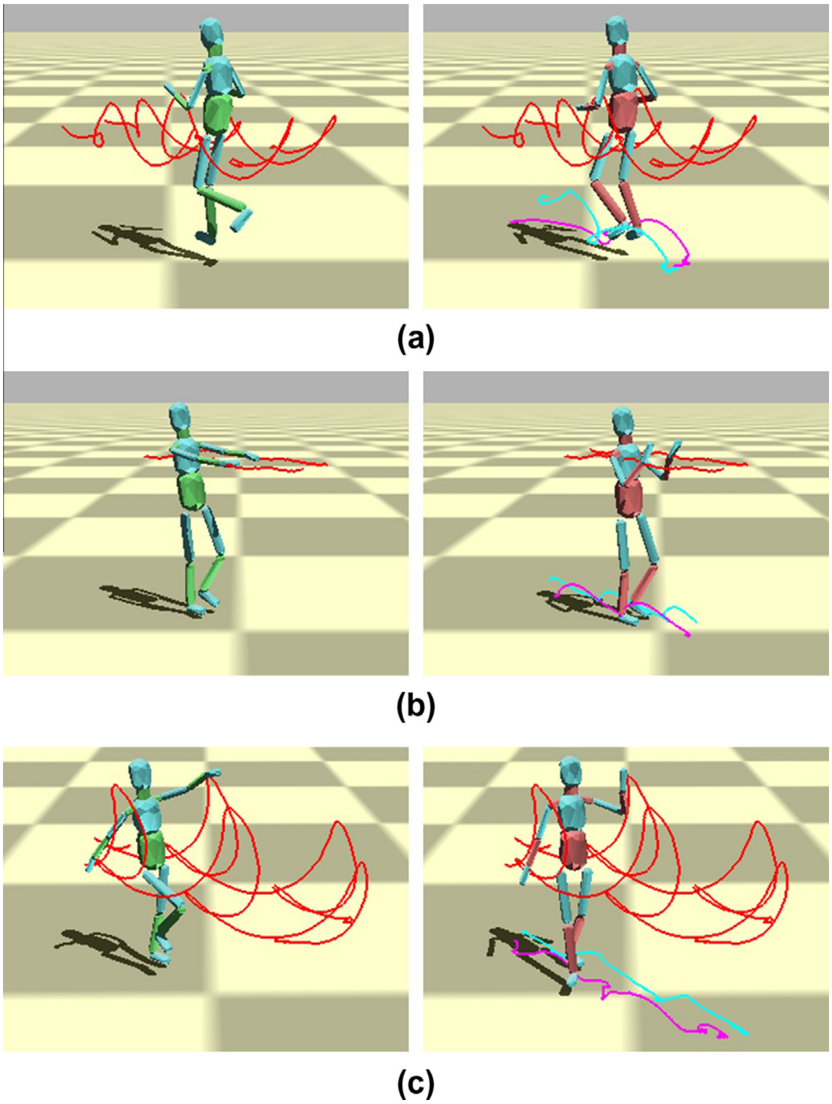


Fig. 18. Examples that reconstructed motions (right) are significantly different from the ground truth input motions (left). (a) Input: hopping twice. Output: jumping and hopping. (b) Input: walking with both arms stretched out. Output: walking with both arms raised comfortably. (c) Input: marching with wide arm swing. Output: funny walking.

joint angle for the test motions in Figs. 14 and 15. Static motions (#4, #8, and #10) tend to show less error and a jumping motion (#2) has the largest error. The graph confirms that reconstruction error is significantly reduced (approximately 32% on average) when ankle trajectories are estimated by our method.

Fig. 17 reports the effect of the noise level on the reconstruction error. The average RMS error per joint angle tends to increase more or less linearly with the standard deviation of the Gaussian noise. Reconstructed motions with varying noise levels are demonstrated in the accompanying video.

We compare the motions reconstructed by our method with those by [8,10] (refer to the accompanying video). In this experiment, we focus on the reconstruction quality of the lower body because all the methods follow the input wrist points well. Because each algorithm uses a different method for estimating the pelvis motion, we remove the effect of the pelvis estimation by applying the ground truth pelvis motion to all methods. In addition, instead of accelerometer data, we use wrist positions as the input signal for testing the method of [10].

As the method of [8] was developed for the case that both ankle and wrist points are given as inputs, it produces unnatural lower-body motions, such as foot-skating and sudden movement change, if ankle trajectories are not used. By improving the temporal coherency in constructing the local linear model, Tautges et al. [10] reconstructs whole-body motions successfully when the input wrist motion is sufficiently discriminative to find a single behavior. However, if two or more behaviors are found from one input, the algorithm may create discrete changes to other behaviors, as shown in the accompanying video. Our method also finds multiple behaviors in this case, but the ankle trajectory generation process avoids the discrete changes in the behaviors.

The compute time for generating the optimal ankle trajectories is approximately 100 miliseconds per frame without using the fast algorithm to find K_1 examples such as kd-trees. The compute time for the whole-body reconstruction given the ankle and wrist trajectories is approximately 350 miliseconds per frame.

8. Limitations and future work

Because we reconstruct a whole-body motion only from the wrist trajectories, different behaviors of similar wrist trajectory patterns cannot be distinguished. For example, using the wrist trajectory for the hopping motion, a jumping motion can be reconstructed, as shown in Fig. 18(a) and (b) shows another example that a different style of motion with similar wrist trajectories is reconstructed given an input motion that is unlike any motions in the database. Even in that case, our method still creates a feasible motion due to the ankle trajectory generation algorithm.

Like any other data-driven methods that searches for example poses from a motion database, the range of motions created by our method depends on that of the database. An apparent limitation is that the proposed method may not reconstruct natural motions successfully if motions with similar wrist patterns are not included in the database. Fig. 18(c) shows such an example that a funny walking motion that has not been included in the motion database is synthesized for input wrist trajectories of a marching motion.

In finding the optimal ankle trajectories, the sliding term heuristically increases the naturalness of the motion by penalizing foot skating, which casts a limitation that our method cannot accurately reconstruct sliding motions. It remains a challenge to develop a more principled method to quantify the naturalness of ankle trajectories so that a wider range of input motions can be reconstructed with enhanced realism.

There are a number of interesting venues for future work. Because the ankle trajectory generation is performed per some time interval, temporal delay would be somewhat unavoidable and it may impede the interactivity for real-time applications. The delay could be mitigated to some extent by adopting techniques for efficient computation, such as constructing a DAG online and parallelizing processes for finding the shortest path in the DAG.

We plan to develop techniques for using two mobile devices to track the three-dimensional trajectories of the wrist points to capture the performance of a human. This will greatly reduce the cost and effort required to capture everyday activities.

Incorporating environmental information into the performance capture process is another interesting aspect of future work. By recognizing the geometric and semantic information of an environment, such as the existence of obstacles and the function of an object, it will be possible to reconstruct whole-body motion more accurately with a minimal number of sensors.

9. Conclusion

In this paper, we have presented a novel data-driven method to synthesize whole-body animation using the trajectories of two wrist positions. To this end, we developed a technique to estimate the ankle trajectories from the input wrist trajectories by solving for the shortest-path problem in a DAG. Requiring minimal attachment of sensors or markers, the proposed method offers a convenient method for capturing whole-body motion able to be recognized from an arm movement. We demonstrated the validity and effectiveness of the method by reconstructing various motions, such as walking, jogging, jumping, and turning sharply, as well as in-place upper-body motions.

Acknowledgments

This work was supported by the Global Frontier R&D Program funded by NRF, MSIP, Korea (2012M3A6A 3055690), and by the Basic Science Research Program of NRF, Korea (2010-0025725). We wish to thank Sukwon Lee for implementing a footskate cleanup algorithm.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.gmod.2013.08.002.

References

[1] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, J. Popović, Practical motion capture in everyday surroundings, ACM Transactions on Graphics 26 (3) (2007) (article 35).

[2] R. Slyper, J.K. Hodgins, Action capture with accelerometers, in: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08), 2008, pp. 193–199.

[3] T. Shiratori, J.K. Hodgins, Accelerometer-based user interfaces for the control of a physically simulated character, ACM Transactions on Graphics 27 (5) (2008) 123:1–123:9.

[4] T. Shiratori, H.S. Park, L. Sigal, Y. sheikh, J.K. Hodgins, Motion capture from body-mounted cameras, ACM Transactions on Graphics 30 (4) (2011) 31:1–31:10.

[5] C. Rose, M.F. Cohen, B. Bodenheimer, Verbs and adverbs: multidimensional motion interpolation, IEEE Computer Graphics and Applications 18 (5) (1998) 32–40.

[6] L. Kovar, M. Gleicher, F. Pighin, Motion graphs, ACM Transactions on Graphics 21 (3) (2002) 473–482.

[7] K. Grochow, S.L. Martin, A. Hertzmann, Z. Popovi'c, Style-based inverse kinematics, ACM Transactions on Graphics 23 (3) (2004) 522531.

[8] J. Chai, J.K. Hodgins, Performance animation from low-dimensional control signals, ACM Transactions on Graphics 24 (3) (2005) 686–696.

[9] P. Kelly, C.Ó. Conaire, J.K. Hodgins, N.E. O'Connor, Human motion reconstruction using wearable accelerometers (poster), in: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10), 2010.

[10] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H. Seidel, B. Eberhardt, Motion reconstruction using sparse accelerometer data, ACM Transactions on Graphics 30 (3) (2011) 1–12.

[11] H. Liu, X. Wei, J. Chai, I. Ha, T. Rhee, Realtime human motion control with a small number of inertial sensors, in: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 2011), 2011, pp. 133–140.

[12] B. Krüger, J. Tautgest, A. Weber, A. Zinke, Fast local and global similarity searches in large motion capture databases, in:

Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10), 2010, pp. 1–10.
[13] G. Klein, D. Murray, Parallel tracking and mapping for small ar workspace, in: Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR '07), 2007, pp. 13–16.
[14] G. Klein, D. Murray, Parallel tracking and mapping on a camera phone, in: Proceedings of the 8th International Symposium on Mixed and Augmented Reality (ISMAR '09), 2009, pp. 83–86.
[15] J. Lee, S.Y. Shin, A hierarchical approach to interactive motion editing for human-like figures, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99), 1999, pp. 39–48.
[16] J. Lee, S. Shin, A coordinate-invariant approach to multiresolution motion analysis, Graphical Models 63 (2) (2001) 87–105.
[17] O. Arikan, D.A. Forsyth, Interactive motion generation from examples, ACM Transactions on Graphics 22 (3) (2002) 483–490.
[18] K. Pullen, C. Bregler, Motion capture assisted animation: texturing and synthesis, ACM Transactions on Graphics 21 (3) (2002) 501–508.
[19] L. Ren, G. Shakhnarovich, J.K. Hodgins, H. Pfister, P. Viola, Learning silhouette features for control of human motion, ACM Transactions on Graphics 24 (4) (2005) 1303–1331.
[20] O. Arikan, D.A. Forsyth, J.F. O'Brien, Motion synthesis from annotations, ACM Transactions on Graphics 22 (3) (2003) 402–408.
[21] A. Safonova, J.K. Hodgins, Construction and optimal search of interpolated motion graphs, ACM Transactions on Graphics 26 (3) (2007) 106:1–106:11.
[22] J. Lee, K. Lee, Precomputing avatar behavior from human motion data, Graphical Models 68 (2) (2006) 158–174.
[23] X. Wei, J. Min, J. Chai, Physically valid statistical models for human motion generation, ACM Transactions on Graphics 30 (3) (2011) 19:1–19:10.
[24] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, N.S. Pollard, Interactive control of avatars animated with human motion data, in: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02), 2002, pp. 491–500.
[25] T. Mukai, S. Kuriyama, Geostatical motion interpolation, ACM Transactions on Graphics 24 (3) (2005) 1062–1070.
[26] L. Kovar, M. Gleicher, Automated extraction and parameterization of motions in large data sets, ACM Transactions on Graphics 23 (3) (2004) 559–568.
[27] L. Zhao, A. Safonova, Achieving good connectivity in motion graphs, Graphical Models 71 (4) (2009) 139–152.
[28] C. Bregler, S. Omohundro, Nonlinear image interpolation using manifold learning, Advances in Neural Information Processing Systems (1994) 973–980.
[29] Y. Li, T. Wang, H.-Y. Shum, Motion texture: a two-level statistical model for character motion synthesis, ACM Transactions on Graphics 21 (3) (2002) 465–472.
[30] M. Brand, A. Hertzmann, Style machines, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), 2000, pp. 183-192.
[31] S. Levine, J.M. Wang, A. Haraux, Z. Popović, V. Koltun, Continuous character control with low-dimensional embeddings, ACM Transactions on Graphics 31 (4) (2012) 28:1–28:10.
[32] J.M. Wang, D.J. Fleet, A. Hertzmann, Gaussian process dynamical models for human motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2) (2008) 283298.
[33] Y. Ye, C.K. Liu, Synthesis of responsive motion using a dynamic model, Computer Graphics Forum 29 (2) (2010) 555562.
[34] N.D. Lawrence, Probabilistic non-linear principal component analysis with gaussian process latent variable models, Journal of Machine Learning Research 6 (2005) 1783–1816.
[35] J. Min, Y.-L. Chen, J. Chai, Interactive generation of human animation with deformable motion models, ACM Transactions on Graphics 29 (1) (2009) 9:1–9:12.
[36] J. Min, J. Chai, Motion graphs++: a compact model for semantic motion analysis and synthesis, ACM Transactions on Graphics 31 (6) (2012) 153:1–153:12.
[37] R. Heck, L. Kovar, M. Gleicher, Splicing upper-body actions with locomotion, Computer Graphics Forum 25 (3) (2006) 459–466.
[38] K. Yin, D.K. Pai, FootSee: an interactive animation system, in: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03), 2003, pp. 329–338.
[39] L. Kovar, M. Gleicher, J. Schreiner, Footskate cleanup for motion capture editing, in: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02), 2002, pp. 97–104.
[40] M.I.A. Lourakis, Levmar: Levenberg–Marquardt nonlinear least squares algorithms in C/C++, <http://www.ics.forth.gr/~lourakis/levmar/>, 2004.
[41] R.G. Brown, Smoothing, Forecasting and Prediction of Discrete Time Series, Courier Dover Publications, 2004.