Research Article

International Journal of Distributed Sensor Networks

International Journal of Distributed Sensor Networks
2017, Vol. 13(2)

DOI: 10.1177/1550147717696083
journals.sagepub.com/home/ijdsn
SAGE

Dance motion capture and composition using multiple RGB and depth sensors

Yejin Kim

Abstract

Dynamic human movements such as dance are difficult to capture without using external markers due to the high complexity of a dancer's body. This article introduces a marker-free motion capture and composition system for dance motion that uses multiple RGB and depth sensors. Our motion capture system utilizes a set of high-speed RGB and depth sensors to generate skeletal motion data from an expert dancer. During the motion acquisition process, a skeleton tracking method based on a particle filter is provided to estimate the motion parameters for each frame from a sequence of color images and depth features retrieved from the sensors. The expert motion data become archived in a database. The authoring methods in our composition system automate most of the motion editing processes for general users by providing an online motion search with an input posture and then performing motion synthesis on an arbitrary motion path. Using the proposed system, we demonstrate that various dance performances can be composed in an intuitive and efficient way on client devices such as tablets and kiosk PCs.

Keywords

Motion capture, dance motion, motion acquisition, motion composition, motion authoring

Date received: 14 September 2016; accepted: 3 February 2017

Academic Editor: Janez Perš

Introduction

Performing arts such as dance and theater are in the domain of intangible cultural heritage that risk becoming buried in oblivion if not archived in a proper record. It is particularly difficult to convey the expressive content of dance to an audience without showing bodily movements explicitly. To digitize the full-body motion of a dancer, optical motion capture systems[1] are commonly used due to their high accuracy in generating the motion data by tracking a set of external markers. However, these markers, attached to the body, often restraint the dancer's dynamic movements and affect the performances during the motion capture session.

However, marker-free systems can produce motion data without using markers or the required tightly fitted suits. Recently, the availability of off-the-shelf depth sensors such as Microsoft Kinect[2] has drawn much attention as a way to track and capture human motion in real time. While systems with a single Kinect sensor suffer from the body's occlusion and rotation problems,[3] systems with multiple sensors can capture a dancer's posture at different angles and combine these views into continuous motion data.[4–7]. However, these approaches have mainly targeted the capturing of relatively simple steps and dance gestures, which are not suitable for complicated and dynamic movements in ballet, modern, and K-pop dances.

School of Games, Hongik University, Sejong, Republic of Korea

Corresponding author:
Yejin Kim, School of Games, Hongik University, Sejong 30019, Republic of Korea.
Email: yejkim@hongik.ac.kr

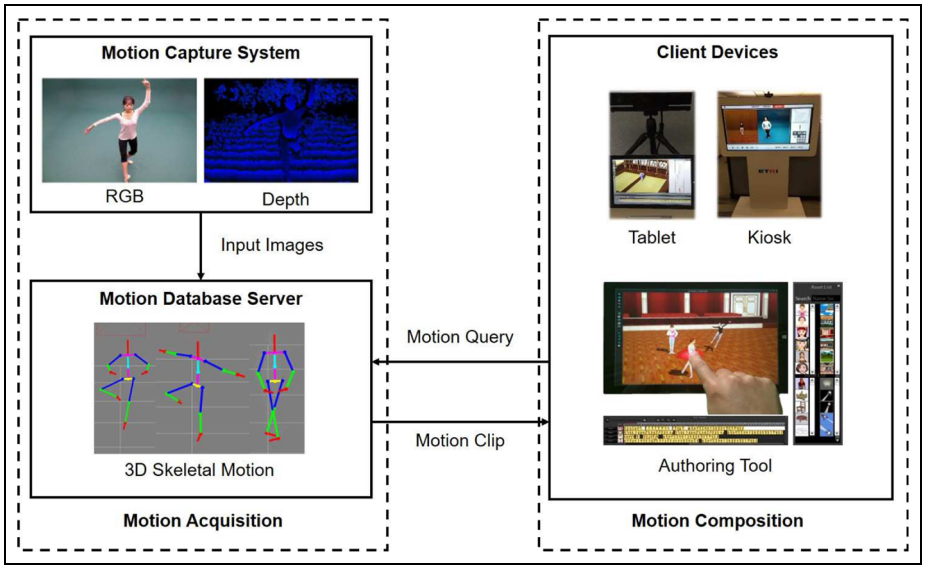

Figure 1. System overview of dance motion acquisition and composition.

In this article, we propose a marker-free motion capture and composition system that can generate motion capture data from expert dancers and compose new dance performances for general users using the captured data. Figure 1 shows an overview of our system in two processes: motion acquisition and motion composition. During the motion acquisition process, our capture system tracks the bodily movements of an expert dancer based on the input data, RGB and depth (RGB-D) images retrieved from multiple RGB-D sensors and constructs a sequence of three-dimensional (3D) skeletal postures from the input data. By archiving the expert motion data as a database in the main server, our system provides an authoring tool for general users to compose various dance performances on the client devices such as tablets and kiosk PCs. During this composition process, the user can search for a specific motion clip throughout the motion database using one's own posture and synthesize continuous motion by displacing the searched motion clips on a specified path.

Our system makes two main contributions. First, we provide a marker-free system that combines off-the-shelf color and depth sensors to capture full-body dance motions. To do this, a particle filter–based method with raw sensor data (RGB-D images) is introduced to track the 3D skeletal postures of an expert dancer. In addition, we present an iterative closest point (ICP)-based method for unifying the skeleton data retrieved from multiple sensors. Second, with the expert motion data, our system can be applied to compose various dance performances for a general user. We provide authoring methods for intuitive motion composition: an online motion search with a user posture and a motion synthesis of multiple motion clips on a specified motion path, which can aid the user's composition activities on the client devices.

The rest of this article is organized as follows. Previous approaches for human motion capture with marker-free systems are reviewed in section "Related work." The motion acquisition process from an expert dancer and the motion composition process for a general user are detailed in sections "Motion acquisition" and "Motion composition," respectively. The experimental results are demonstrated in section "Experimental results." We conclude this article with a discussion of future improvements in section "Conclusion."

Related work

Over the years, human motion capture has been actively studied by many researchers, especially in the computer vision field. Human motion acquisition based on vision techniques is well surveyed by Moeslund et al.[8] and Chen et al.[9] Without using external markers, much of the vision-based approaches can be grouped into three categories: generative (also known as *top-down*), *discriminative* (also known as *bottom-up*), and *hybrid*.

The generative approaches[10–14] usually rely on an external model of the human body and try to estimate the model parameters that best describe the pose in an input image. Aguiar et al.[10] used a highly detailed 3D model to deform mesh appearances by estimating the 3D correspondence on multi-view images. Similarly, Gall et al.[11] produced both skeletal motion and mesh deformation by fitting the template model onto the silhouettes extracted from multi-view images.

Ganapathi et al.[12] introduced a real-time system that tracks human motion from a sequence of depth images based on the probabilistic temporal model. In their approach, a set of physical constraints is used to deform the simplified body model. Using a Gaussian mixture model without establishing a point correspondence between the template model and the subject, Ye and Yang[13] tracked skeletal motion and a rough mesh model of articulated objects in real time. With two synchronized RGB-D sensors, Michel et al.[14] adopted a stochastic optimization technique to track the skeletal motion from the depth volume. However, the prerequisite template model and its initialization for model parameters have made these approaches difficult for capturing different types of dance movements without additional data.

However, the discriminative approaches[15–18] try to identify body parts directly from the input images via a learning process. Michoud et al.[15] introduced a 3D shape estimation method that tracks body motions based on the silhouettes segmented from the multi-view images. In their approach, the tracking performance depends on the size of the image set used for the segmentation. Given a labeled training set of image patches, Plagemann et al.[16] used local shape descriptors to detect the salient body parts only. Shotton et al.[17] estimated the 3D joint positions from a single depth image by training the randomized decision forest classifier with a large image set. Recently, Jung et al. achieved a large performance gain for 3D human pose estimation by training a regression tree for each joint.

Backed by an existing database, the hybrid approaches[19–23] try to improve the tracking accuracy by complementing the generative methods (i.e. the optimization problems) with the discriminative methods (i.e. the database reference). Ganapathi et al.[19] developed an interactive system that detects body parts throughout a kinematic chain in a stochastic framework. Combined with a randomized decision forest classifier,[17] Wei et al.[21] tracked skeletal motion in real time. This work was further improved by Zhang et al.[23] with the use of additional sensor data. Baak et al.[20] performed an extensive use of database references to search for similar poses with the salient body detection.[16] Later, Helten et al.[22] present a similar approach with a personalized tracker that can estimate various body shapes. However, both the discriminative and hybrid approaches require a large database in advance of the tracking process, which is not suitable for capturing various motion types from different dancers.

Recently, the availability of an off-the-shelf sensor such as Microsoft Kinect[2] makes it possible to capture real-time human motions in a cost-effective way. As a single use of Kinect can suffer from self-occlusion and body rotation problems,[3] multiple Kinects can be used to maximize the tracking performance all around a dancer. Berger et al.[4] and Zhang et al.[5] utilized multiple Kinects for posture estimation. However, their methods targeted non-skeletal motion data. For dance motion, Kitsikidis et al.[6] adopted a hidden conditional random fields (HCRF) classifier to recognize motion patterns fused from multi-Kinects. Baek and Kim[7] presented a similar approach for combining the postures, which included mixing the five joint segments tracked by the multi-Kinects system. However, the dance movements in these approaches are slow and simple, while our system mainly targets more dynamic motions such as ballet, modern, and K-pop dances.

When dance motions are given as a set of example data, Fan et al.[24] established a relationship between the input music and motions based on the dynamic programming method and synthesis dance motions that are synchronized in music. Panagiotakis et al.[25] synthesized a new sequence of motion patterns from the periodic examples using a motion graph. Unlike these approaches, our system focuses on synthesizing continuous motions by displacing a set of short motion clips on an arbitrary motion path.

Motion acquisition

System setup

To capture dynamic motion from an expert dancer without using external markers, our system consists of multiple high-speed RGB[26] and time-of-flight (ToF) depth[27] sensors. As shown in Figure 2, a set of RGB-D sensors is configured at hexagonal positions in a green-walled studio with light-emitting diode (LED) lights. These sensors cover 360° of the dancer and reduce the noises incurred from the background. In addition, this sensor configuration minimizes the well-known interference problem caused by the infrared emissions from multiple ToF sensors.[28] To capture the input images at

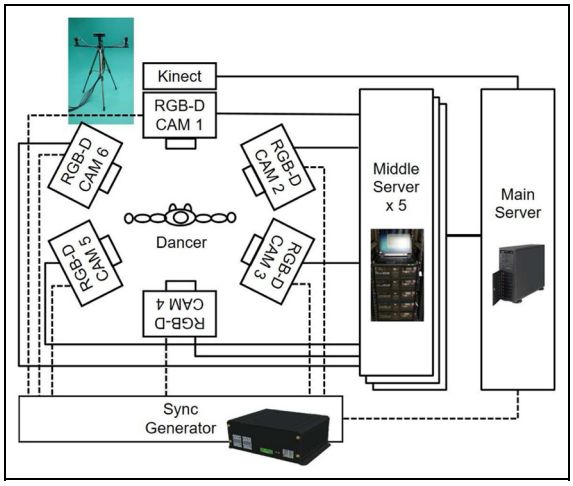

Figure 2. Overview of motion capture system with multiple RGB-D sensors.

Table 1. Sensors used for the motion acquisition and composition.

	RGB (Tracking)	Depth (Tracking)	Depth (Posture)
Model	 CREAVIS GigE Vision	 MESA SR4000	 Microsoft Kinect v2
Resolution	640 × 480	176 × 144	512 × 424
Frame Rate (fps)	120	30	30
Type	CCD	Time-of-Flight	Time-of-Flight
Interface	Ethernet Base-T	Ethernet Base-T	USB 3.0
Ext. Trigger	yes	yes	No

the same time, an external sync generator controlled by the main server via a triggering signal is used for the sensor synchronization. Due to the large data bandwidth required for the continuous streams of uncompressed images (i.e. RAW format) from the multiple sensors, a set of middle servers is connected between the sensors and the main server. Each of these servers connects either two RGB or three depth sensors, while the main server gathers and processes the input data from those middle servers. It is noteworthy that a separate Kinect sensor[2] is connected to the main server and used to obtain an initial skeleton model of the dancer to be tracked. Table 1 shows three different sensors used in our system.

Skeletal motion tracking

The proposed system generates 3D skeletal motions by tracking the input data, which are a set of RGB-D images retrieved from multiple RGB-D sensors. At first, the tracking process is initialized by registering an articulated skeleton model (initial joint positions) retrieved from the Kinect sensor,[2] which is controlled by the main server. This skeleton model consists of 21 internal and end joints in a hierarchical structure as shown in Figure 3. Due to the limited reconstruction of skeleton postures from a single depth image,[17] the dancer should face toward the sensor with all the joints visible (i.e. a T-pose or an N-pose) during the skeleton registration.

Provided with the skeleton model $\mathbf{S}$, our system tracks its motion $\mathbf{S}_t$ by estimating a global position of the ith joint J_t^i at discrete time t such that $\mathbf{S}_t = [J_t^1, J_t^2, \ldots, J_t^{21}]$ and $J_t^i \in \mathbb{R}^3$. For this, our tracking process adopts a particle filter–based method[29] to estimate J_t^i from a color cue image $\mathbf{c}_t$ and a depth cue image $\mathbf{d}_t$ as follows

$$p(\mathbf{S}_t|\mathbf{c}_{t-1}, \mathbf{d}_{t-1}) = \int p(\mathbf{S}_t|\mathbf{S}_{t-1}) p(\mathbf{S}_{t-1}|\mathbf{c}_{t-1}, \mathbf{d}_{t-1}) d\mathbf{S}_{t-1} \quad (1)$$

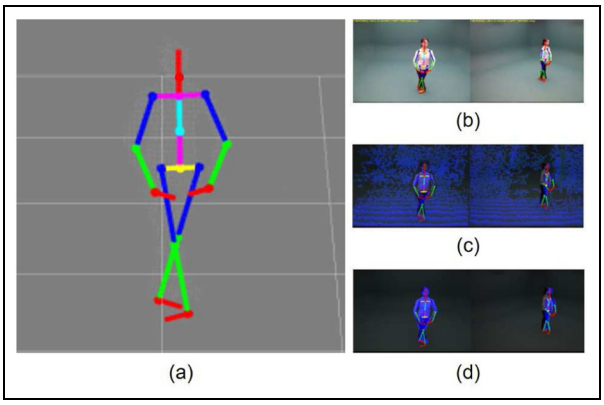

Figure 3. Skeletal motion tracking at different angles: (a) a hierarchical skeleton model used for output, (b) RGB input images, (c) depth input images, and (d) filtered RGB-D input images.

as a prediction step and

$$p(\mathbf{S}_t|\mathbf{c}_t, \mathbf{d}_t) \propto p(\mathbf{c}_t, \mathbf{d}_t|\mathbf{S}_t) p(\mathbf{S}_t|\mathbf{c}_{t-1}, \mathbf{d}_{t-1}) \quad (2)$$

as a filtering step. Here, a Gaussian sampling process with the prior function $p(\mathbf{S}_t|\mathbf{S}_{t-1})$ generates a number of particles in $\mathbf{S}_t$. This sampling is used to approximate the likelihood of the current observation $p(\mathbf{c}_t, \mathbf{d}_t|\mathbf{S}_t)$, which is defined as

$$p(\mathbf{c}_t, \mathbf{d}_t|\mathbf{S}_t) \propto \left(- \sum_{\mathbf{c} \in \{R,G,B\}} \frac{D_{\mathrm{B}}^2(\mathbf{c}_t, \mathbf{c}_{t-1})}{2\sigma_{\mathbf{c}}^2} - \sum_{\mathbf{d} \in \{N_d\}} \frac{D_{\mathrm{B}}^2(\mathbf{d}_t, \mathbf{d}_{t-1})}{2\sigma_{\mathbf{d}}^2} \right)^{\frac{1}{2}} \quad (3)$$

where $D_{\mathrm{B}}(\cdot)$ measures the Bhattacharyya distance[30] between the two cues at t and $t-1$, and N_d is the number of depth samples. The particles of $\mathbf{c}_t$ and $\mathbf{d}_t$ are sampled from the averaged RGB-D values in the small circles between two linked joints on the RGB-D images, respectively. As shown in Figure 3, the RGB-D images with the subtracted background are used for precise particle-based motion tracking.

Skeletal motion unification

As shown in Figure 4, visible joints and their tracked positions are different from one sensor to another over time. At each frame, our system unifies them into one skeletal motion by transforming each of the skeletons to a reference coordinate system and selecting a joint that has a minimum positional difference between t and $t-1$.

When one of the RGB-D sensors is selected for a reference skeleton $\mathbf{S}_t^R$, we can estimate the rigid transformation matrix $\mathbf{M}_T$ that transforms the coordinate system in $\mathbf{S}_t$ of each sensor to the reference one in $\mathbf{S}_t^R$. Based on the observation of dancers' tendency to face

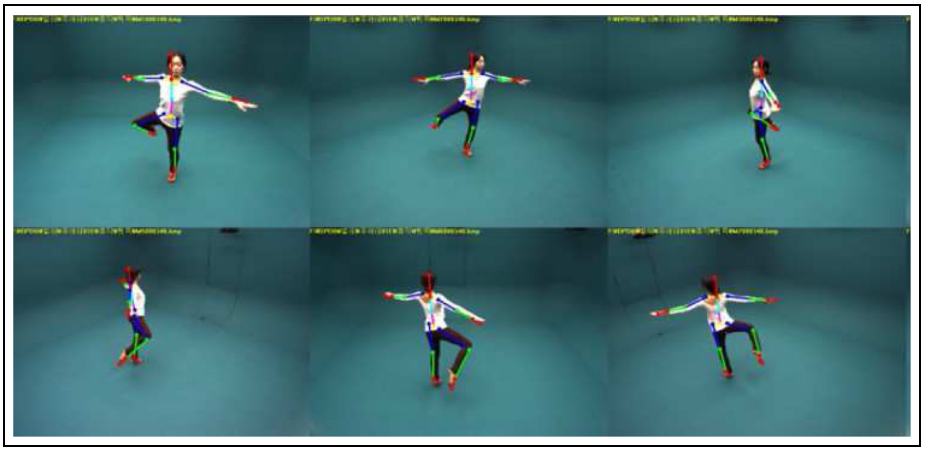

Figure 4. 3D skeletal motion tracked by multiple RGB-D sensors.

toward the front sensor (i.e. RGB-D Cam 1 in Figure 2) during their performances, this sensor is used for $\mathbf{S}_t^R$ and remains in the fixed position during an entire capturing session. To derive $\mathbf{M}_T$, our system utilizes the ICP method[31] due to its computational efficiency and monotonic convergence. However, the unified skeleton from ICP can be erroneous due to the sparse number of matching points such as the joints in our skeleton model. For this reason, a point cloud P_t is generated from $\mathbf{d}_t$ of each sensor by tessellating the depth space in $\mathbf{d}_t$ with N_p voxels and estimating an average position p_j in each voxel, where $j \in [1, \ldots, N_p]$. Similarly, if there is a reference point cloud P_t^R generated from $\mathbf{d}_t^R$, $\mathbf{M}_T$ for each sensor can be estimated by minimizing the following error function

$$E(\mathbf{M}_T) = E(\mathbf{R}, \mathbf{T}) \propto \frac{1}{N_p} \sum_{j=1}^{N_p} \| p_j^R - \mathbf{R}_N(\mathbf{R}_{pj} + \mathbf{T}) \|^2 \quad (4)$$

where $p_j^R \in P_t^R$ and $p_j \in P_t$ sampled from $\mathbf{d}_t^R$ and $\mathbf{d}_t$, respectively. Here

$$\mathbf{T} = \bar{p}^R - \mathbf{R}\bar{p} \quad (5)$$

where

$$\bar{p}^R = \frac{1}{N_p} \sum_{j=1}^{N_p} p_j^R \text{ and } \bar{p} = \frac{1}{N_p} \sum_{j=1}^{N_p} p_j \quad (6)$$

Given a correlation matrix $\mathbf{W}$

$$\mathbf{W} = \sum_{j=1}^{N_p} \hat{p}_j^R \hat{p}_j^T = \mathbf{U}\mathbf{C}\mathbf{V}^T \quad (7)$$

where $\hat{p}_j^R = p_j^R - \bar{p}^R$ and $\hat{p}_j = p_j - \bar{p}$. Thus, the optimal solution for $E(\mathbf{M}_T)$ is $\mathbf{R} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{W} = \mathbf{U}\mathbf{C}\mathbf{V}^T$ derived from a single value decomposition (SVD).

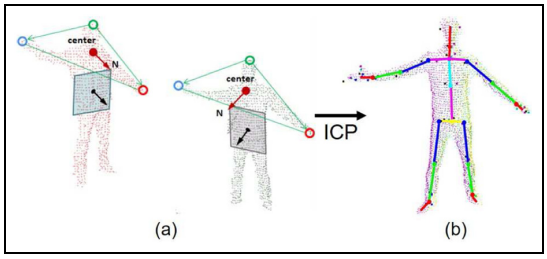

Figure 5. Skeleton unification: (a) two point clouds with different body orientations and (b) the unified skeleton model via the ICP method.

As shown in Figure 5, $\mathbf{R}_N$ is the body rotation between P_t and P_t^R,which is estimated by a plane composed of three points from the head and two hand regions in P_t. This alignment of two point clouds accelerates the iterative process in ICP.

Given $\mathbf{M}_T$ and $\mathbf{S}_t$ of each sensor, the unified skeleton $\overline{\mathbf{S}}_t$ can be constructed as follows

$$\overline{\mathbf{S}}_t = [\overline{J}_t^1, \overline{J}_t^2, \ldots, \overline{J}_t^{21}] \quad (8)$$

where $\overline{J}_t^i$ is the ith joint in $\overline{\mathbf{S}}_t$ and estimated by averaging the corresponding joint positions in $\mathbf{M}_T^k \mathbf{S}_t^k$, where $k \in [1, \ldots, N_k]$ and N_k is the number of sensors used in the system. To remove noisy joints, we discard any joint in $\mathbf{M}_T^k \mathbf{S}_t^k$ that has the Euclidean difference, measured between J_{t-1}^i and J_t^i, outside the threshold value set by a user sampled from the ground-truth data. Here, an optimal estimator such as Kalman filter can be used as an alternate solution for posture reconstruction; however, its smoothing effect can potentially subside some of high-frequency details in human motions.[32]

Motion database

The skeletal motion $\overline{\mathbf{S}}_t$ consists of a set of joint positions moving over time t. However, this representation

is not suitable for animating the rigid body model with fixed skeleton lengths for the dance composition. For this reason, the proposed system converts $\overline{\mathbf{S}}_t$ to a rotational presentation using inverse kinematics[33] as follows

$$\widehat{\mathbf{S}}_t = [p_t^1, r_t^1, r_t^2, \ldots, r_t^{21}] \tag{9}$$

where p_t^1 is the global position of the root joint and r_t^i is the rotation of the ith joint with respect to its parent, which is represented by a unit quaternion.[34]

A fixed length of the bone segment $l^{i-1,i}$ between $\overline{J}_t^{i-1}$ and $\overline{J}_t^i$ can be estimated by solving a least-squares problem for each bone segment as follows

$$\underset{l^{i-1,i}}{\arg\min} = \sum_{t=1}^{N_t} \| \overline{J}_t^i - (\overline{J}_t^{i-1} + \mathbf{d}^{i-1,i} * l^{i-1,i}) \| \tag{10}$$

where $\mathbf{d}^{i-1,i}$ is a normalized direction vector between $\bar{J}_t^{i-1}$ and $\bar{J}_t^i$, and N_t is the total number of frames tracked by the sensors. Here, the new position of the ith joint is expressed in terms of $\mathbf{d}^{i-1,i}$ to simplify the estimation.

Motion Composition

Figure 6 shows an overview of motion composition on client devices such as tablets and kiosk PCs. As the size of the motion database in the main server grows quickly by capturing the dance movements with high-speed RGB-D sensors, it becomes a time-consuming task for a general user to browse the entire database and search for desired motion data. Furthermore, each motion data in the database contains a fixed motion path, requiring the user to edit it on an input path for motion composition. Our system eases this difficulty by providing two authoring methods: *online motion search* and *motion synthesis*.

Online motion search

To search for a specific motion clip $\widetilde{\mathbf{S}}_t$, where $\widetilde{\mathbf{S}}_t \in \overline{\mathbf{S}}_t$, a user posture $\mathbf{S}^U$ is retrieved by the Kinect v2 sensor[2] from the client devices and used as an input query in the motion database, as shown in Figure 6. Due to the high complexity of the articulated skeleton model in $\mathbf{S}^U$, a direct comparison between $\mathbf{S}^U$ and the large number of frames in $\overline{\mathbf{S}}_t$ is not only inefficient but also has too many redundant results.[35] For this reason, our search method defines a set of feature vectors $\mathbf{v}_i^U$ for $\mathbf{S}^U$ and $\mathbf{v}_{i,t}^D$ for $\overline{\mathbf{S}}_t$, respectively, where $i \in [1, \ldots, N_\mathbf{v}]$ and $N_\mathbf{v}$ is the number of feature vectors used for the comparison. In our system, we set $N_\mathbf{v} = 5$ such that each $\mathbf{v}_i$ is a normalized Euclidean distance from a root joint (i.e. *SpineBase* in $\mathbf{S}^U$) to one of the end-effector joints (i.e. *HandRight*, *HandLeft*, *AnkleRight*, *AnkleLeft*, and *Head* in $\mathbf{S}^U$), respectively. In addition, the body orientations $\mathbf{n}^U$ for $\mathbf{S}^U$ and $\mathbf{n}_t^D$ for $\overline{\mathbf{S}}_t$ are defined by estimating two normals $\mathbf{n}_1$ and $\mathbf{n}_2$ from the plane configured by three torso joints (i.e. *SpineMid-HipRight-HipLeft* and *SpineMid-ShoulderRight-ShoulderLeft* in $\mathbf{S}^U$), as shown in Figure 7. Similarly, $\mathbf{v}_{i,t}^D$ and $\mathbf{n}_t^D$ are defined from the corresponding joints in $\overline{\mathbf{S}}_t$, respectively.

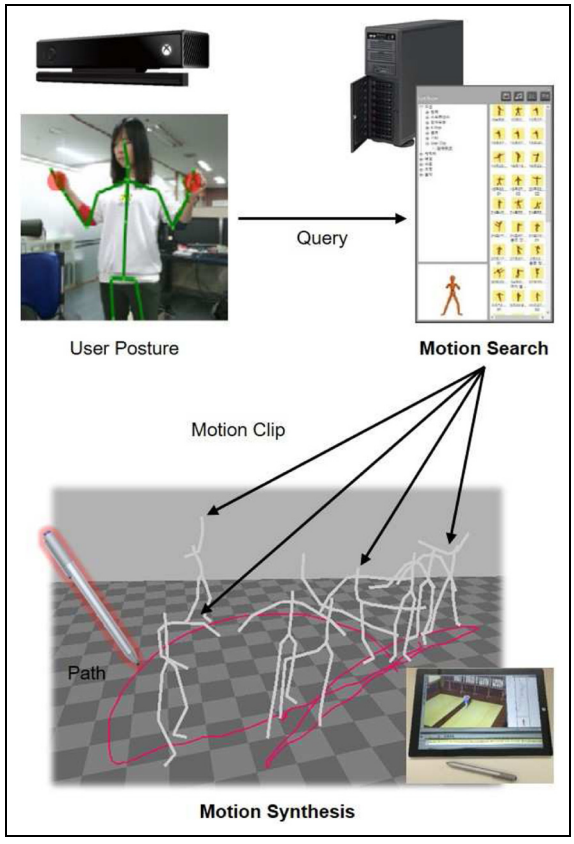

Figure 6. Overview of motion composition on a client device.

Given $\mathbf{v}_i$ and $\mathbf{n}_{1,2}$ between $\mathbf{S}^U$ and each posture in $\overline{\mathbf{S}}_t$, their similarity $S(\cdot)$ is measured as follows

$$S(\mathbf{S}^U, \overline{\mathbf{S}}_t) = \min\left(D_\mathbf{v}\left(\mathbf{v}_i^U, \mathbf{v}_{i,t}^D\right) + D_\mathbf{n}\left(\mathbf{n}^U, \mathbf{n}_t^D\right)\right) \tag{11}$$

where $D_\mathbf{v}(\cdot) = \sum_{i=1}^{N_\mathbf{v}} w_i^\mathbf{v} \| \mathbf{v}_i^U - T_{\theta,\bar{x},\bar{z}}\mathbf{v}_{i,t}^D \|^2$ and $D_\mathbf{n}(\cdot)$ is the weighted angular difference between N^U and N_t^D, which can be easily estimated via the dot product. Here, $w_i^\mathbf{v}$ is a weight value that scales the importance of $\mathbf{v}_i$ during the search process. If more precise comparison is needed, $T_{\theta,\bar{x},\bar{z}}$ is used to align the two postures, which rotates $\mathbf{v}_{i,t}^D$ about the vertical (y) axis by θ degrees and is then translated by $(\bar{x}, \bar{z})$.[36]

A searched motion clip, $\widetilde{\mathbf{S}}_t$, is a sequence of N_f frames defined by a user. Using a window size of N_f frames with weights w_j^S, where $j \in [1, \ldots, N_f]$, our search method retrieves $\widetilde{\mathbf{S}}_t$ from $\overline{\mathbf{S}}_t$ based on the sum of $w_j^S S(\cdot)$. Here, w_j^S tapers off from the center $N_f/2$ to both ends of the window, 1 and N_f, respectively.

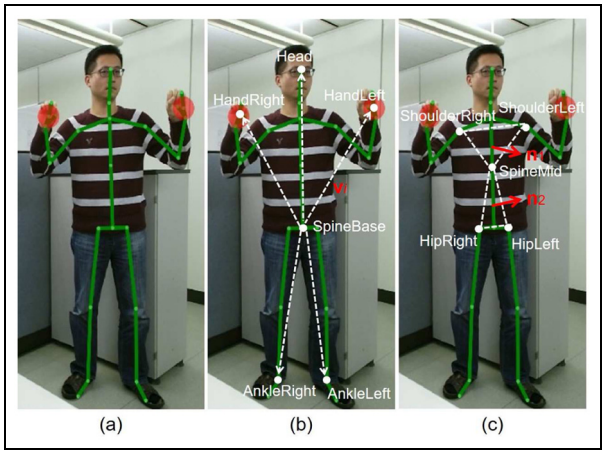

Figure 7. Posture parameters used for online motion search: (a) skeleton structure of a user posture, (b) a set of feature vectors defined from end-effector positions, and (c) normal vectors defined for body orientations.

Motion synthesis

As shown in Figure 6, when an arbitrary path is drawn on the ground (i.e. x–z) plane using an input device (i.e. a digital pen or a finger tip) on the client screen, our synthesis method concatenates multiple motion clips into one continuous motion on the path. A continuous and smooth path can be generated by fitting an interpolation spline $\widehat{\mathbf{P}}$ such as a cubic B-spline with $N_c + 1$ control points to a rough path $\mathbf{P}$ drawn by a user. The optimal number of N_c is determined by

$$N_c = \left\lceil \sum_{i=2}^{N_s} S_f \parallel s_i - s_{i-1} \parallel \right\rceil \tag{12}$$

where s_i is a sample point derived from $\mathbf{P}$ and is smoothed by the one-dimensional (1D) Gaussian filter with a window size of 8. Here, N_s is the total number of input points in $\mathbf{P}$, and S_f is a scaling factor that normalizes $\mathbf{P}$. Thus, $\widehat{\mathbf{P}}$ is a parametric curve defined by $\widehat{\mathbf{P}}(u), u \in [0, \ldots, 1]$. In our method, S_f is the ratio of a user's height and average of step lengths, which is approximately 0.41.

The temporal location of each motion clip on $\widehat{\mathbf{P}}$ is specified by a user via the timing editor, as shown in Figure 8. To generate a continuous and smooth sequence of output motions, the motion blending technique[37] is used to concatenate one motion to another. The gap between two motion clips in the timing editor determines the blending duration between two motion clips. Let $\mathbf{P}^R$ be a curved path of a root joint from the skeleton model in each clip, which is projected on the ground plane. The new positions and orientations for the root joint on $\widehat{\mathbf{P}}(u)$ are determined from the arc length $\mathbf{A}(u), u \in [0, \ldots, 1]$. Similarly, $\mathbf{A}(u)$ is defined by fitting $\mathbf{P}^R$ to $\widehat{\mathbf{P}}$ with the cubic B-spline curve.

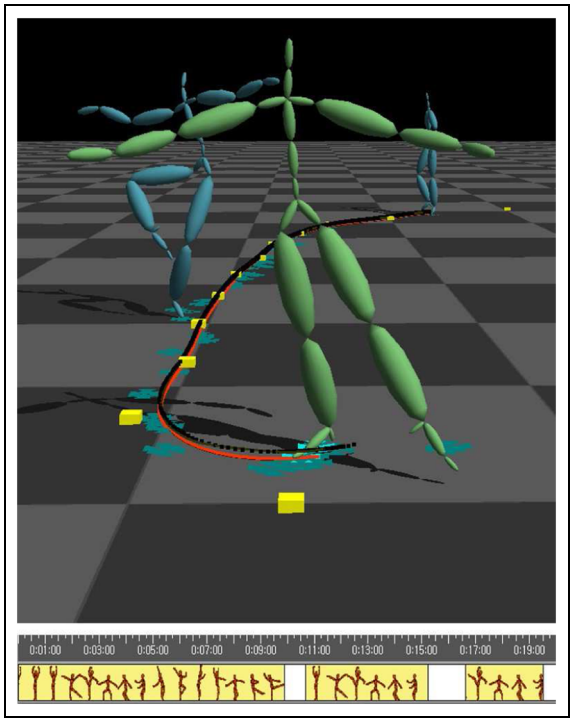

Figure 8. Displacement of multiple motion clips on a specified motion path: three different motion clips are placed on the path via the timing editor shown. Here, the path (red) with a set of control points (yellow) is fitted to the input points (black).

Experimental results

We demonstrated the applicability of our system by capturing various dance movements from expert dancers and providing general users the captured motion data for dance motion composition. As the RGB-D sensors in the system capture 120 and 30 frames per second (fps), respectively, we set $\mathbf{d}_{t-3} = \mathbf{d}_{t-2} = \mathbf{d}_{t-1} = \mathbf{d}_t$ for the motion tracking process. Our system is best understood through examples of its use, as described below, and the accompanying video.

Dance motion capture

As shown in Figure 9, various dance movements are captured from expert dancers in ballet, modern, and Latin dances without using external markers and a special suit. The dancers wear ordinary clothes that they usually use during a practice session, except for the bluish color used for our studio walls. To evaluate the tracking accuracy of dance motion, our system is compared against the multi-Kinects system[7] that consists of four Kinect sensors. At the same time, the ground-truth data are captured by a commercial system[38] that uses a set of inertial sensors embedded in a wearable suit.

Due to the differences in the skeleton structure and size produced between the comparing systems, the

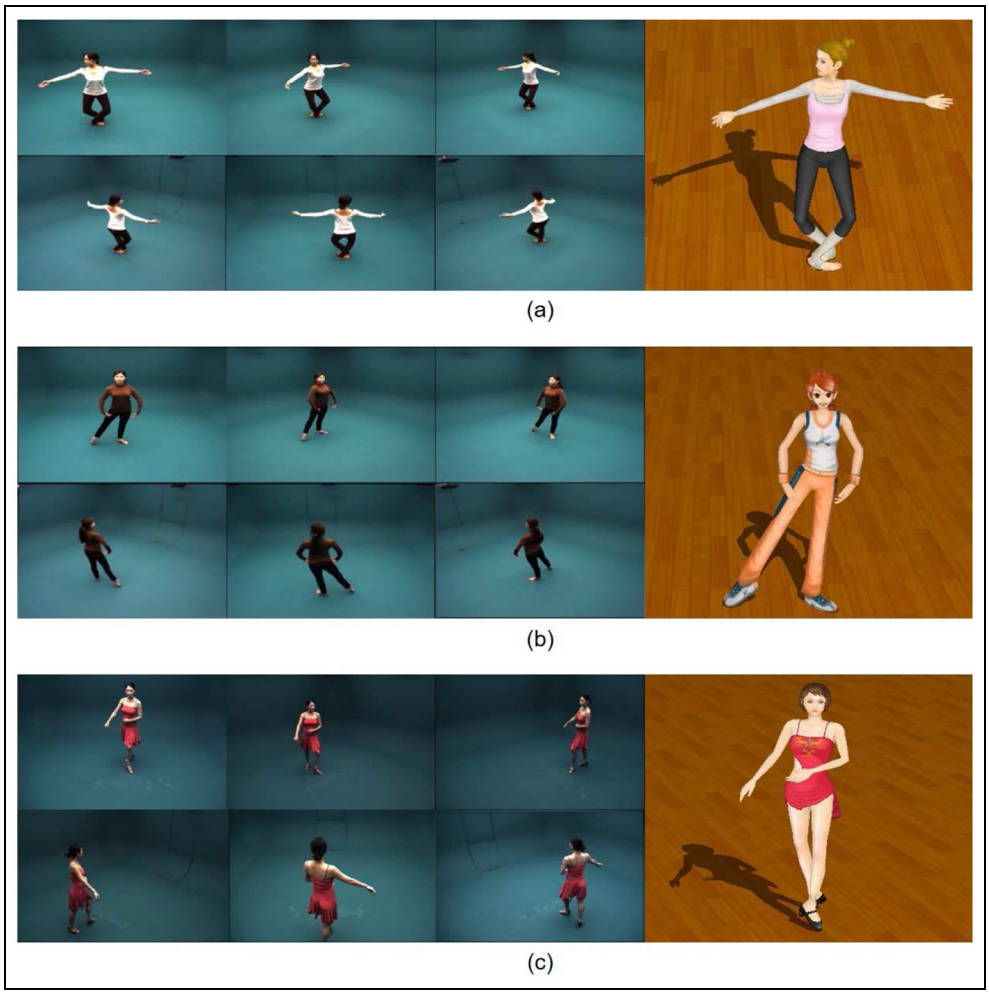

Figure 9. Dance motion capture from (a) a ballet, (b) a modern, and (c) a Latin dancer with a 3D character model.

direct comparison between two skeleton postures is inaccurate for tracking tests. For this reason, we have adopted the online motion retargeting method, which maps the different skeleton structures to template one,[39] and then used the positional differences of joints in the global coordinate as an accuracy measure. Figure 10 compares the tracking accuracy of our system against the multi-Kinects system based on the ground-truth data captured by the commercial system. In this comparison, a total of 30,728 frames (about 256 s) are captured from ballet, modern, and Latin dances which include dynamic movements such as cross steps, rapid turns, stretches, and high jumps. As a result, our system tracks the dance movements at an average of 89.5% for wrist, 82.5% for ankle, and 92.0% for head joints against the ground-truth data, making our system considerably more accurate than the multi-Kinects system (i.e. 69.3% for wrist, 58.8% for ankle, and 74.3% for head joints). It is noticeable that the accuracy of the legs are lower than other parts due to the higher noises around the dancer's feet in the depth images. In addition, the tracking accuracy of the legs in Latin dance is relatively lower than the other dances. This is mainly because parts of the dancer's legs are occluded by the stage costume, which affects the tracking performance during the motion acquisition process. However, the decrease in accuracy is relatively small compared to the multi-Kinects system.

Dance motion composition

For the dance motion database used by a general user, we captured the motion data from ballet, modern, K-pop, Latin, and traditional Korean dancers, respectively. Table 2 shows the total frames and processing time for capturing each type of motion by our system. All the motion data in the database are down-sampled to 30 fps to speed up the online search process with the user's posture.

As shown in Figure 11, our system provides user interfaces that let the user draw a motion path and displace the searched motion clips directly on the device

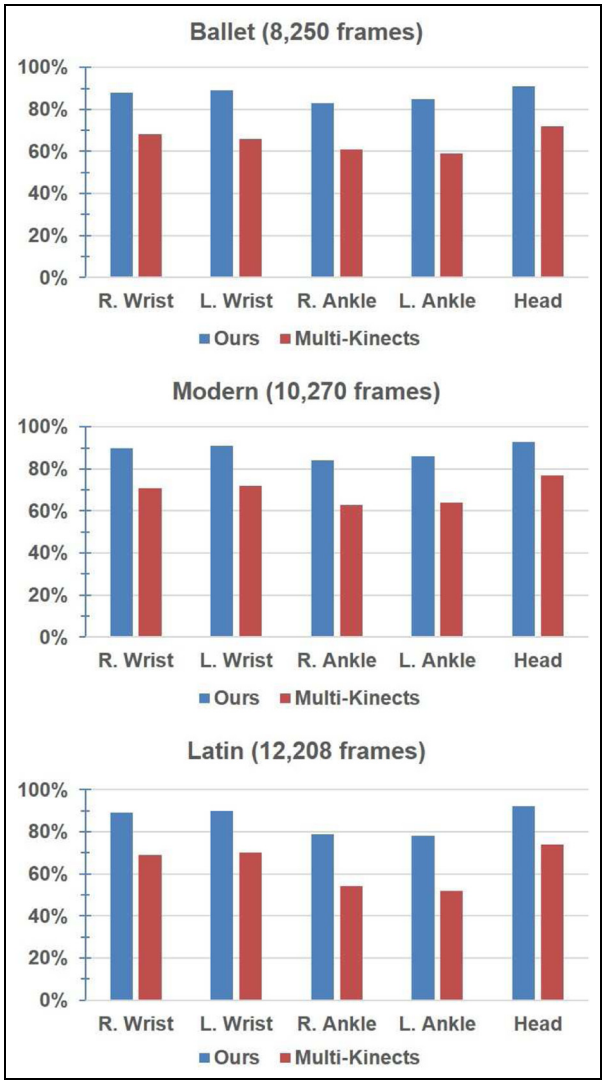

Figure 10. Tracking accuracy of our system and the multi-Kinects system[7] against the commercial system[38] for ballet, modern, and Latin dances.

Table 2. Dance motion database: archived at 30 fps.

Category (types)	Total frames (s)	Process time seconds (fps)
Ballet (25)	255,220 (8507)	24,077 (10.6)
Modern (17)	72,345 (2412)	5742 (12.6)
K-pop (10)	63,681 (2123)	5397 (11.8)
Latin (4)	143,890 (4796)	11,698 (12.3)
Korean (10)	126,726 (4224)	9183 (13.8)

screen. Figure 12 shows an instance of synthesizing four motion clips on the specified path into one continuous motion. Each of the motion clips is searched throughout the database with a length of $N_f = 60$ frames (2 s) using a user posture that is captured from a single Kinect attached to the client device. The searched clips are displayed to the user based on its similarity level

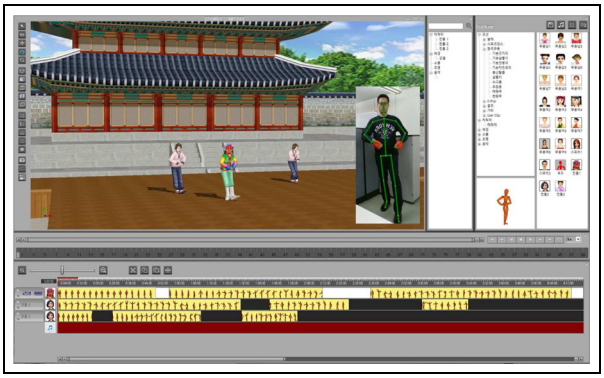

Figure 11. User interfaces for motion composition.

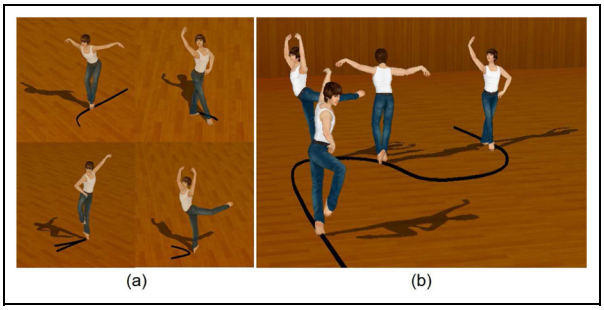

Figure 12. Motion synthesis: (a) searched motion clips with a fixed path and (b) displaced motion clips on a specified path.

measured by equation (11). In our system, the synthesis processes (generating a smooth motion path and blending between two motion clips) only require specifying the timings on the path from the user. Figure 13 shows various dance performances on the virtual stages, which are composed by users who have no experience in the motion-based content production.

Conclusion

In this article, we have introduced a motion capture system that can track dynamic movements in dance motions without using external markers. Based on the RGB-D cues retrieved from multiple RGB-D sensors, our system can generate 3D skeletal motions from various expert dancers. To compose dance performances on virtual stages for general users, our system provides authoring methods that can search a set of desired motion clips from the given motion database and synthesize the stage scene by displacing the motion clips on a specified path. As demonstrated in the experimental results, various dance performances can be produced on the client devices in intuitive and efficient ways. In practice, our system can be used to archive various dance performances into the motion database to be used for theater stage plans, movement education, and ultimately, heritage preservation in dance.

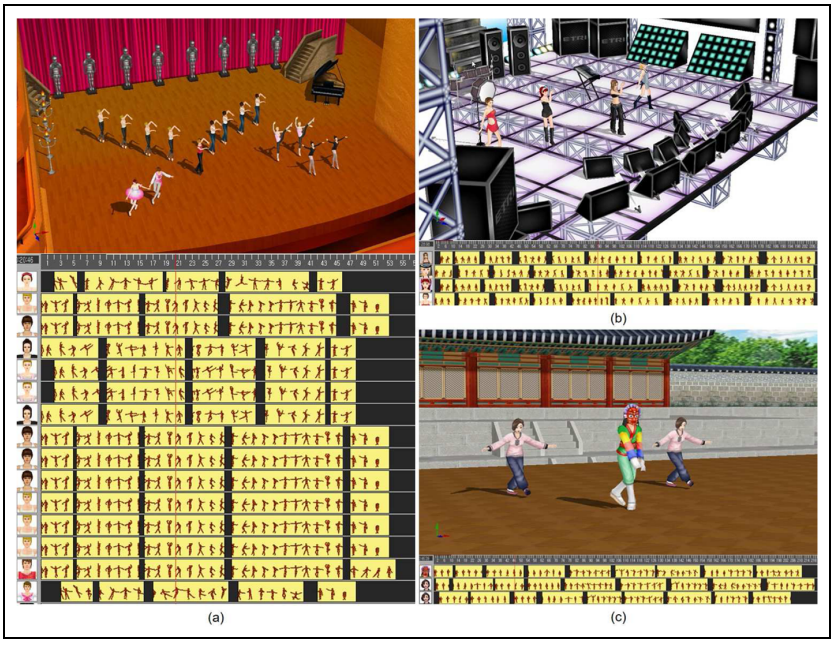

Figure 13. Composition of various dance performances on stages: (a) ballet, (b) K-pop, and (c) traditional Korean dance with corresponding motion clips used.

One of the ongoing improvements in the current system is enhancing the motion quality around the feet area. Due to the ambiguity between the feet and touching the ground, the high noise levels in the depth images degrade the tracking performance of our system. We expect that adding a small and weightless inertial sensor on each foot can obtain more precise joint rotations without affecting the user's freedom of performances. In addition, using an articulated template model with image segmentation techniques can be a potential solution for better motion quality.[40]

The tracking performance of our system is affected by the type of dress. During the motion capture session, it is not unusual that the expert dancer wears a voluminous costume that makes difficult to track the bodily movement inside. We are currently working on the extraction of skeletal postures from such a dress based on the probability model.

The current system requires numerous sensors and system PCs to capture motions, incurring a high system cost. Depending on the dance type and processing time, the system configuration can be scaled down by removing some of the sensors and servers. For example, for a slower and simpler type of dance, three or four RGB-D cameras in optimal capturing positions can generate comparable motion data. The number of servers can be reduced if the processing time for motion data generation is not important for dance composition. Finally, a predefined skeleton template can be used as an initial posture model to save the Kinect sensor in our system.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Hongik University new faculty research support fund.

References

1. Vicon motion capture system, https://www.vicon.com
2. Microsoft Kinect v2 camera, https://developer.microsoft.com
3. Alexiadis D, Kelly P, Daras P, et al. Evaluating a dancer's performance using kinect-based skeleton tracking. In: *Proceedings of the international conference on multimedia*, Scottsdale, AZ, 28 November–1 December 2011, pp.659–662. New York: ACM.
4. Berger K, Ruhl K, Shcroeder Y, et al. Markerless motion capture using multiple color-depth sensors. In: *Proceedings of the vision, modeling and visualization workshop*, Berlin, 4–6 October 2011, pp.317–324. Geneva: Eurographics.
5. Zhang L, Sturm J, Cremers D, et al. Real-time human motion tracking using multiple depth cameras. In: *Proceedings of the international conference on intelligent robots and systems*, Vilamoura-Algarve, 7–12 October 2012, pp.2389–2395. New York: IEEE.
6. Kitsikidis A, Dimitropoulos K, Douka S, et al. Dance analysis using multiple kinect sensors. In: *Proceedings of the international conference on computer vision theory and*

applications, Lisbon, 5–8 January 2014, pp.789–795. New York: IEEE.
7. Baek S and Kim M. Dance experience system using multiple kinects. *Future Comput Commun* 2015; 4(1): 45–49.
8. Moeslund TB, Hilton A and Kruger V. A survey of advances in vision-based human motion capture and analysis. *Comput Vis Image Und* 2006; 104(2–3): 90–126.
9. Chen L, Wei H and Ferryman J. A survey of human motion analysis using depth imagery. *Pattern Recogn Lett* 2013; 23(15): 1995–2006.
10. Aguiar ED, Theobalt C, Stoll C, et al. Marker-less deformable mesh tracking for human shape and motion capture. In: *Proceedings of the conference on computer vision and pattern recognition*, Minneapolis, MN, 17–22 June 2007, pp.1–8. New York: IEEE.
11. Gall J, Stoll C, Aguiar ED, et al. Motion capture using joint skeleton tracking and surface estimation. In: *Proceedings of the conference on computer vision and pattern recognition*, Miami, FL, 20–25 June 2009, pp.1746–1753. New York: IEEE.
12. Ganapathi V, Plagemann C, Koller D, et al. Real-time human pose tracking from range data. In: *Proceedings of the European conference on computer vision*, Firenze, 7–13 October 2012, pp.738–751. Berlin: Springer.
13. Ye M and Yang R. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In: *Proceedings of the conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp.2353–2360. New York: IEEE.
14. Michel D, Panagiotakis C and Argyros AA. Tracking the articulated motion of the human body with two RGBD cameras. *Mach Vision Appl* 2015; 26(1): 41–54.
15. Michoud B, Guillou E, Briceno H, et al. Real-time marker-free motion capture from multiple cameras. In: *Proceedings of the international conference on computer vision*, Rio de Janeiro, Brazil, 14–21 October 2007, pp.1–7. New York: IEEE.
16. Plagemann C, Ganapathi V, Koller D, et al. Real-time identification and localization of body parts from depth images. In: *Proceedings of the international conference on robotics and automation*, Anchorage, AK, 3–8 May 2010.
17. Shotton J, Fitzgibbon A, Cook M, et al. Real-time human pose recognition in parts from a single depth image. In: *Proceedings of the conference on computer vision and pattern recognition*, Providence, RI, 16–21 June 2012, pp.1297–1304. New York: IEEE.
18. Jung HY, Lee J, Heo YS, et al. Random tree walk toward instantaneous 3D human pose estimation. In: *Proceedings of the conference on computer vision and pattern recognition*, Boston, MA, 7–12 June 2015, pp.2467–2474. New York: IEEE.
19. Ganapathi V, Plagemann C, Koller D, et al. Real time motion capture using a single time-of-flight camera. In: *Proceedings of the conference on computer vision and pattern recognition*, San Francisco, CA, 13–18 June 2010.
20. Baak A, Muller M, Bharaj G, et al. A data-driven approach for real-time full body pose reconstruction from a depth camera. In: *Proceedings of the international conference on computer vision,* Barcelona, 6–13 November 2011, pp.1092–1099. New York: IEEE.
21. Wei X, Zhang P and Chai J. Accurate realtime full-body motion capture using a single depth camera. *Trans Gr* 2012; 31(6): 1–12.
22. Helten T, Baak A, Bharaj G, et al. Personalization and evaluation of a real-time depth-based full body tracker. In: *Proceedings of the international conference on 3D vision*, Seattle, WA, 29 June–1 July 2013, pp.279–286. New York: IEEE.
23. Zhang P, Siu K, Zhang J, et al. Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *Trans Gr* 2014; 33(6): 1–14.
24. Fan R, Xu S and Geng W. Example-based automatic music-driven conventional dance motion synthesis. *IEEE T Vis Comput Gr* 2012; 18(3): 501–515.
25. Panagiotakis C, Argyros A and Michel D. Temporal segmentation and seamless stitching of motion patterns for synthesizing novel animations of periodic dances. In: *Proceedings of the international conference on pattern recognition*, Stockholm, 24–28 August 2014, pp.1892–1897. New York: IEEE.
26. CREAVIS GigE Vision RGB camera, http://www.crevis.co.kr
27. Mesa Imaging SR4000 ToF camera, http://hptg.com/industrial
28. Bhandari A, Kadambi A, Whyte R, et al. Resolving multi-path interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Opt Lett* 2014; 39(6): 1705–1708.
29. Doucet A, Freitas N and Gordon N. Sequential Monte Carlo methods in practice. New York: Springer, 2001.
30. Aherne F, Thacker N and Rockett P. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika* 1997; 3(4): 1–7.
31. Besl PJ and McKay ND. A method for registration of 3-D shapes. *IEEE T Pattern Anal* 1992; 14(2): 239–256.
32. Sul C, Jung S and Wohn K. Synthesis of human motion using Kalman filter. In: Magnenat-Thalmann N and Thalmann D (eds) *Modelling and motion capture techniques for virtual environment*. Berlin: Springer, 1998, pp.100–112.
33. Zhao J and Badler N. Inverse kinematics positioning using nonlinear programming for highly articulated figure. *ACM T Graphic* 1994; 13: 313–336.
34. Shoemake K. Animating rotation with quaternion curves. In: *Proceedings of the ACM SIGGRAPH computer graphics*, San Francisco, CA, 22–26 July 1985, pp.245–254. New York: ACM.
35. Keogh E, Palpanas T, Zordan VB, et al. Indexing large human-motion databases. In: *Proceedings of the 13th international conference on very large data bases*, Toronto, ON, Canada, 31 August–3 September 2004, pp.780–791. VLDB Endowment Inc.
36. Kovar L, Gleicher M and Pighin F. Motion graphs. *ACM Trans Gr* 2002; 21(3): 473–482.
37. Kim Y and Neff M. Automating expressive locomotion generation. *Trans Edutain VII* 2012; 7145: 46–61.
38. Xsens MVN motion capture system, http://xsens.com
39. Choi K-J and Ko H-S. On-line motion retargetting. *J Visual Comp Animat* 2000; 11(5): 223–235.
40. Liu Y, Gall J, Stoll C, et al. Markerless motion capture of multiple characters using multiview image segmentation. *IEEE T Pattern Anal* 2013; 35(11): 2720–2735.