Motion Capture of the Human Body Using Multiple Depth Sensors

Yejin Kim, Seongmin Baek, and Byung-Chull Bae

The movements of the human body are difficult to capture owing to the complexity of the three-dimensional skeleton model and occlusion problems. In this paper, we propose a motion capture system that tracks dynamic human motions in real time. Without using external markers, the proposed system adopts multiple depth sensors (Microsoft Kinect) to overcome the occlusion and body rotation problems. To combine the joint data retrieved from the multiple sensors, our calibration process samples a point cloud from depth images and unifies the coordinate systems in point clouds into a single coordinate system via the iterative closest point method. Using noisy skeletal data from sensors, a posture reconstruction method is introduced to estimate the optimal joint positions for consistent motion generation. Based on the high tracking accuracy of the proposed system, we demonstrate that our system is applicable to various motion-based training programs in dance and Taekwondo.

Keywords: Motion capture, Human motion, Dynamic movements, Depth sensor, Multiple Kinect sensors, Training contents.

I. Introduction

Digitizing human motion has long been explored and is characterized by applicability in diverse fields, such as use of virtual agents in gaming, film, animation, and the sports industry. To retrieve full-body motion from a performer, marker-based systems, such as optical systems [1]–[2], are widely used owing to their high accuracy of reconstructing skeletal data from a large set of markers attached to the performer's body. However, these markers, along with a specially designed suit, are difficult to apply to a general user, that is, a novice in the respective dance form, sport, martial art, and so on. The applicability of such systems is thus limited to offline applications.

Marker-free systems, on the other hand, can obtain human motion data without markers or suit wearing. These motion-capture approaches [3]–[7] rely on the three-dimensional (3D) shape and color analysis, or a 3D body model, from a sequence of multi-view images obtained from multiple color cameras. Recently, the emergence of off-the-shelf depth sensors, such as Microsoft Kinect [8], has attracted considerable attention in tracking and capturing human motion in real time. The Kinect sensor has been applied to many commercial games because it is inexpensive and can extract full-body motions from a general user. Based on the simultaneously captured color image and depth data, this sensor type reconstructs a user's postures in real time, which enables interaction with game contents by employing the user's body movements as input. Other types of depth-sensor-based systems for capturing human motion are well explained in a survey conducted by Chen and others [9].

In this paper, we propose a marker-free motion capture system that can track in real time the fast and dynamic movements of a general user. As shown in Fig. 1, the proposed system generates a user's full-body motion with multiple

Manuscript received Aug. 15, 2016; revised Jan. 11, 2017; accepted Feb. 9, 2017. This research was supported by the Sports Promotion Fund of Seoul Olympic Sports Promotion Foundation from Ministry of Culture, Sports and Tourism (s072016122016).

Yejin Kim (corresponding author, yejkim@hongik.ac.kr) and Byung-Chull Bae (byuc@hongik.ac.kr) are with the School of Games, Hongik University, Seoul, Rep. of Korea.

Seongmin Baek (baeksm@etri.re.kr) is with the SW & Content Research Laboratory, ETRI, Daejeon Rep. of Korea.

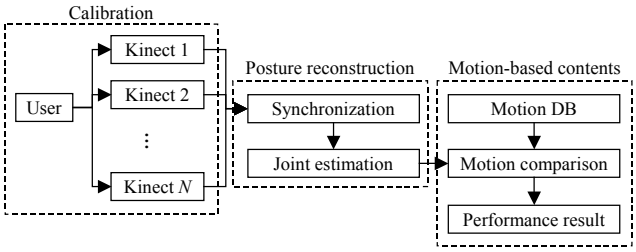


Fig. 1. Overview of motion capture with multiple depth sensors (Kinect v2) and its applicability to motion-based contents.

Kinect sensors via two processes: *calibration* and *posture reconstruction*. During the calibration process, a set of point clouds is sampled from the depth images captured by each sensor. The coordinate systems of the point clouds are unified to comprise a single skeleton representation by using an iterative closest point (ICP) method. During posture reconstruction, an output skeleton posture is reconstructed per frame by estimating the optimal joint positions from a set of noisy skeleton data retrieved from the synchronized sensors. For general users, the applicability of our multi-Kinect system is tested with motion-based content. In these programs, expert's dance motions are captured in advance in a motion database and then compared to the user's motions captured in real time.

Our system makes two main contributions. First, we provide a cost-effective system that utilizes off-the-shelf depth sensors to generate full-body human motion. To this end, we provide calibration and joint estimation methods to overcome joint occlusion and body rotation problems, which are often problematic with a single Kinect sensor. Second, based on the high tracking accuracy of the proposed system, we demonstrate that our multi-Kinect system can be applied to various motion-based content, such as dance and Taekwondo, for training purposes. As shown by the experimental results, dynamic movements in this content can be captured in real time without requiring additional devices or a complicated probability model.

The remainder of this paper is organized as follows. A brief overview of previous approaches to human motion capture with multiple sensors is provided in Section II. The overall system with a server–client model is described in Section III. Calibration of the input data from the Kinect sensors is explained in Section IV. Posture reconstruction for output motion by estimating optimal joints from noisy skeleton data is detailed in Section V. We present the experimental results in Section VI and conclude the paper in Section VII with a discussion of potential improvements.

II. Related Work

Human motion capture without external markers has long been investigated in the computer vision field. Most studies [3]–[7] have focused on using multiple color cameras to capture a user's skeletal movements or surface appearances. Some works [3]–[4] adopt shape and color analysis on a multi-view image sequence to derive the skeletal motion. However, this approach often requires a large set of training images captured from the user. Others [5]–[7] utilize a highly detailed 3D human model for human motion capture. However, these prerequisite data are difficult to prepare in advance for a general user.

Recently, the emergence of inexpensive depth sensors, such as Microsoft Kinect [8], has enabled capturing human motion in real time. However, a system using a single Kinect sensor [10] often suffers from joint occlusion and body orientation problems. When body parts are blocked (for example, when a user turns sideways), the depth data cannot be retrieved from the sensor, leading to unsuccessful posture estimation. For this reason, a user should face the sensor with all joints visible, which is not suitable for dynamic movements performed in dance and martial arts. Izadi and others [11] overcome the occlusion problem by moving the Kinect sensor around a target object and fusing the postures into a single model based on graphics processing unit acceleration. However, their approach is mainly used for static objects.

To address the issue of occlusion of some body parts, several studies adopted use of multiple Kinect sensors to minimize self-occlusion of body parts. Auvinet and others [12] proposed a method for generating 3D body shapes based on visual hulls. Berger and others [13] proposed a method for using four Kinect sensors to extract silhouettes and capture motions. Zhang and others [14] applied particle filtering and partition sampling techniques to track postures. However, these methods draw upon non-skeletal data and an optimization process with silhouette or template matching to estimate postures.

In addition, Williamson and others [15] proposed a soldier training system using multiple Kinect sensors that can capture motion, even when a user rotates 360°. Asteriadis and others [16] proposed a method for applying energy functions to estimate joint positions. Kitsikidis and others [17] used three Kinect sensors to retrieve slow dance motion and the hidden conditional random fields classifier to recognize motion patterns. Kaenchan and others [18] analyzed walking motions based on the mean positions of tracked joints. Moon and others [19] used the Kalman filtering method to alter and mix accurate Kinect data. Nevertheless, Moon and others failed to capture 360° motion events, and the result motions were too simple.

Meanwhile, Jo and others [20] proposed a system using multiple Kinect sensors to track multiple users. However, they focused on tracking the positions of multiple users instead of

retrieving their motions. Ahmed [21] employed four Kinect sensors to capture boxing and walking motions from all around the user. The system tracks the user's face to determine the center sensor. Joint inputs from the other sensors are used to retrieve the skeleton joints that the center one fails to track. Similarly, Baek and Kim [22] assigned a center Kinect sensor based on movements of root joints. They retrieved the postures by mixing the five tracked joint segments. Unlike previous approaches, our system captures more complicated and dynamic movements of general users for training purposes.

III. System Overview

Our proposed system adopts multiple Kinect sensors for motion capture. A single sensor can be connected to a single personal computer (PC) [8]. Thus, N Kinect sensors are required to connect to N PCs. Figure 2 shows one instance of installing two Kinect sensors on each of the four sides (front, rear, left, and right). To process the data inputs from each Kinect sensor, the server–client model is used, whereby all clients are connected to the server PC via an Ethernet connection. During the calibration and posture reconstruction processes, each Kinect PC sends the motion data (that is, a point cloud and skeleton joints) to the server PC, which generates unified skeleton data as a result. These output data are further used by the motion-based content.

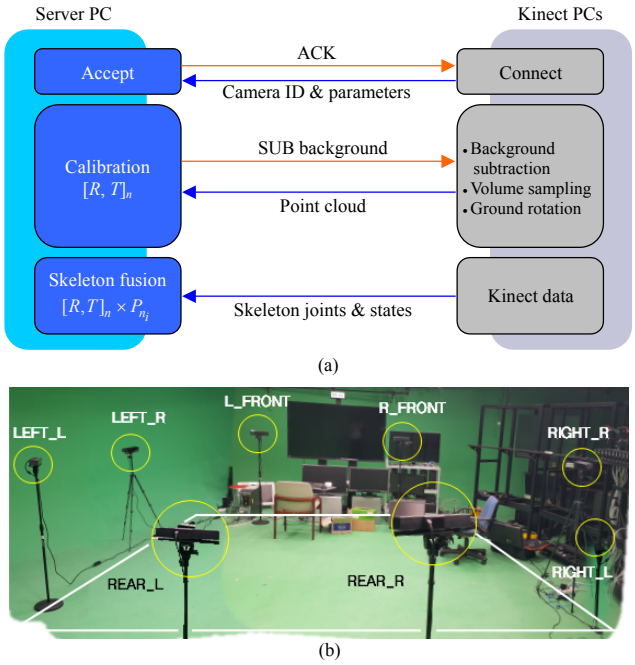


Fig. 2. Overview of the capture studio: (a) server–client model for motion data processes and (b) system configuration with multiple Kinect sensors.

IV. Calibration

As shown in Fig. 2, the user motion data are captured in all directions with multiple Kinect sensors. The coordinate systems of the data inputs from each Kinect sensor differ; therefore, they must be unified into a single coordinate system. To this end, we utilize the ICP method [23] owing to its computational efficiency and monotonic convergence. However, the unified result from ICP can be erroneous because of the sparse number of input data points, such as the joint positions directly retrieved from the Kinect sensor. For this reason, a point cloud of a user is generated from the depth image via background subtraction and volume sampling of the dense depth points.

1. Point Cloud Generation

For the background subtraction, the depth image composed of w by h pixels, which comprise background B_{wh}, is retrieved from Kinect. Once Kinect senses a user with depth image U_{wh}, it compares the depth value of the U_{wh} pixel with that of the saved B_{wh} pixel in the background. When the former is below threshold t_B, it is deemed the background and therefore excluded. In this filtering process, the depth points tend to show noise at the edges. The depth values of eight pixels adjacent to the U_{wh} pixel are compared to determine the similarity of the depth values. When the depth similarity is below a threshold value, t_N, it is considered noise and thus excluded. Finally, as the floor around the user still includes the noise owing to the depth data, the depth data below the positional data of a user's foot-end joint are considered noise and thus excluded.

To generate a point cloud, P^k, from the depth image retrieved from the kth Kinect sensor with the background subtracted, we perform a volume sampling method that tessellates the depth space with N_v voxels and estimates an average position, v_i, in each voxel, where $i \in [1, \ldots, N_v]$. As the physical viewing direction of the kth sensor can be tilted, P^k generated from the depth image should be rotated to align with the ground normal, as shown in Fig. 3. The rotation matrix is determined between the vector associated with the *SpineBase* and *SpineMid* joints of the Kinect skeleton, as shown in Fig. 4 and the up-vector (y-axis) on the ground while the user stands in an upright position. In our system, P^k yields up to approximately 15,000 points with the background and noise eliminated.

2. Coordinate Alignment

With multiple Kinect sensors, the coordinate system of P^k differs for each sensor and can be unified into a single coordinate system by calculating a rigid transformation

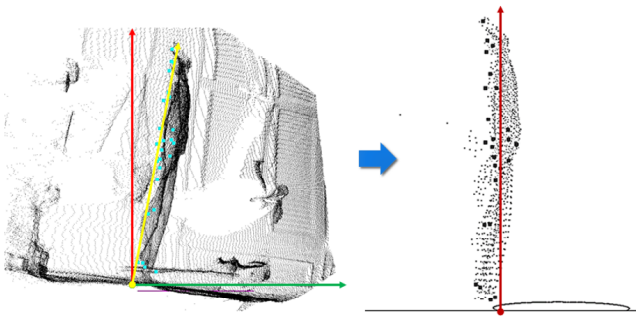


Fig. 3. Point cloud of a user is generated from the depth image after background subtraction and volume sampling of the dense depth points. Here, the point cloud is rotated to align with the ground normal (red) as the sensor can be tilted (yellow).

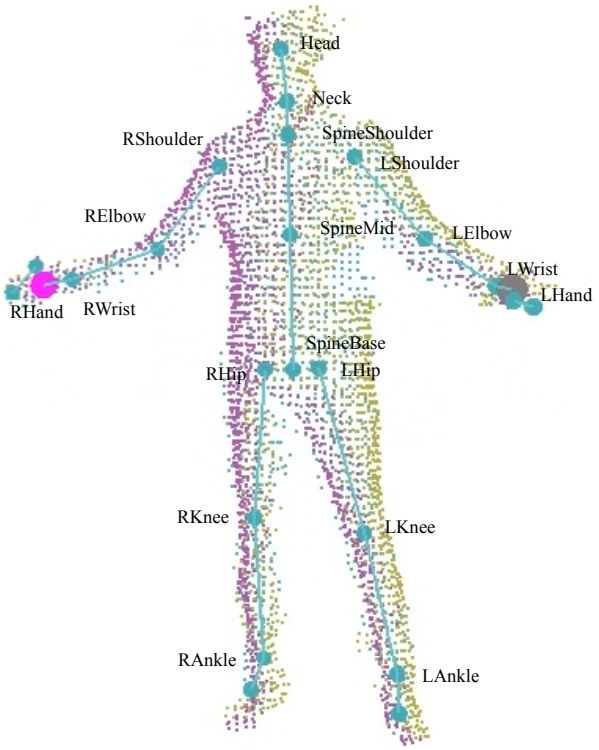


Fig. 4. Skeleton structure used in the posture reconstruction.

$M = [R, T]$ by using the ICP method [23]. Here, R is a rotation and T is a translation of the coordinate system in P^k to the reference coordinate system in P^r, respectively. In our system, the reference point cloud P^r is generated from the depth image retrieved from the front sensor. For efficient calibration, as shown in Fig. 5, a long thin wand (approximately 50 cm) with a light cubic object (for recognition) at its tip is used as a calibration tool. Based on the resolution of the Kinect sensor (521 × 424 pixels), the depth data of the wand are ignored, while the depth data of the tip object are captured. When a user moves the tool in the capture space, the mean value of the

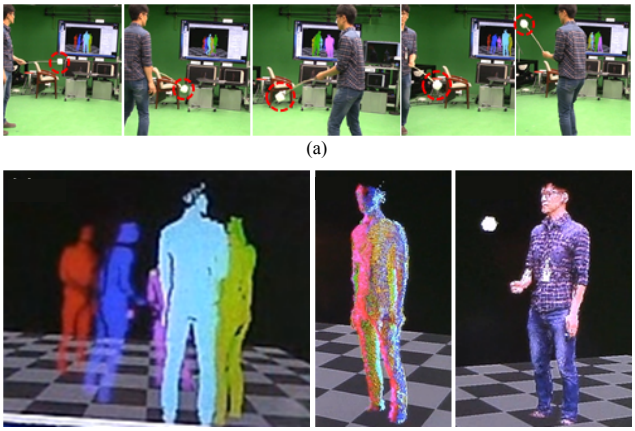


Fig. 5. Calibration process: (a) acquisition of input data points for ICP with a calibration tool, (b) depth images from different Kinect sensors, (c) unified point cloud after transformation, and (d) unified point cloud with color mapping.

depth data of the tip object is saved as the reference point, p_t, where $t \in [1, \ldots, N_p]$. We are provided p_t^k from P^k and p_t^r from P^r and can thus derive M for each Kinect sensor by minimizing the following error function:

$$E(R,T) \propto \frac{1}{N_p}\sum_{t=1}^{N_p}\left\| p_t^r - R_N\left(Rp_t^k + T\right)\right\|^2. \tag{1}$$

Here,

$$T = \bar{p}^r - R\bar{p}^k, \tag{2}$$

where

$$\bar{p}^r = \frac{1}{N_p}\sum_{t=1}^{N_p} p_t^r \text{ and } \bar{p}^k = \frac{1}{N_p}\sum_{t=1}^{N_p} p_t^k. \tag{3}$$

Given correlation matrix W,

$$\begin{aligned} W &= \sum_{t=1}^{N_p} p_t'^r\, p_t'^{kT} \\ &= UCV^T, \end{aligned} \tag{4}$$

where $p_t'^r = p_t^r - \bar{p}^r$ and $p_t'^k = p_t^k - \bar{p}^k$. Thus, the optimal solution for $E(R, T)$ is $R = UV^T$ with $W = UCV^T$ from a single value deposition.

In (1), R_N is the body rotation between P^k and P^r, which is obtained from the normal vectors of the planes. These planes are defined by applying the least squares fitting method [23] on a point cloud as

$$z = Ax + By + C, \tag{5}$$

$$\begin{bmatrix} \sum_{i=1}^{N_v} x_i^2 & \sum_{i=1}^{N_v} x_i y_i & \sum_{i=1}^{N_v} x_i \\ \sum_{i=1}^{N_v} x_i y_i & \sum_{i=1}^{N_v} y_i^2 & \sum_{i=1}^{N_v} y \\ \sum_{i=1}^{N_v} x_i & \sum_{i=1}^{N_v} y_i & \sum_{i=1}^{N_v} 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_v} x_i z_i \\ \sum_{i=1}^{N_v} y_i z_i \\ \sum_{i=1}^{N_v} z_i \end{bmatrix}, \tag{6}$$

where $v_i = \{x_i, y_i, z_i\}$. This alignment of two point clouds accelerates the iterative process in the ICP method. In our system, we set $N_p = 300$, which is collected at an interval of 50 ms.

V. Posture Reconstruction

The tracking performance of a Kinect sensor for skeleton joints is influenced by a user's direction facing toward the sensor. For example, when the user's arms are lifted forward, the sensor tracks the elbow and wrist joints of one arm in the incorrect positions, even when the Kinect tracking state is in the *Not Tracked* setting.

In addition, when a user turns sideways, the sensor no longer can track the joints owing to self-occlusion. Figure 6 shows examples of the self-occlusion problem. In addition, the persistence of the Kinect software (Kinect SDK v2) in continually tracking of hidden joint positions in incorrect positions accumulates tracking errors. However, significant noise can occur when joint values are simply added to determine the mean positions [22]. In our system, a skeleton posture is reconstructed at each frame by estimating the optimal joint positions from the noisy data retrieved from the multiple Kinect sensors.

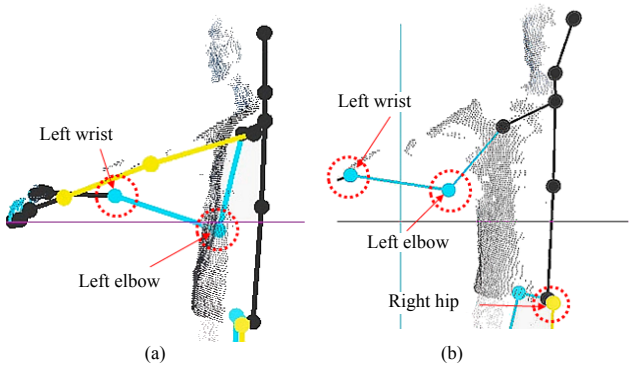


Fig. 6. Examples of skeleton tracking errors from Kinect: (a) forward lift of arms and (b) right turn. Here, the black joints are tracked in correct positions by the sensor, while the blue joints should be in the yellow positions.

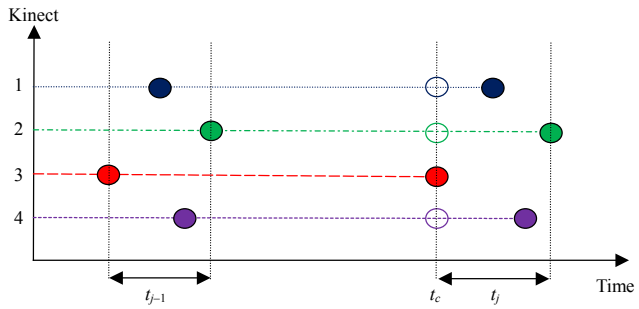


Fig. 7. Synchronization of capture times of multiple Kinect sensors.

1. Synchronization

In general, the Kinect sensor is not designed to work in multiple units; moreover, it cannot simultaneously capture human motion [13]. When each sensor captures a sequence of motion frames, the time difference t_j occurs by dozens of milliseconds (30 ms to 90 ms) from one sensor to another, as shown in Fig. 7. This can be problematic in estimating the optimal joint positions from the joint positions retrieved at each Kinect sensor in t_j.

In our system, a simple method of spline-based interpolation is used to correct the positional differences of the joints captured in t_j. Let t_c be the fastest capture time returned from one of the sensors. The joint positions at t_c for other sensors are estimated by the spline-based interpolation of the previous time t_{j-1} and the current time t_j. Here, the cubic splines, such as Hermite or Ferguson, can be applied between the unit interval of t_{j-1} and t_j.

2. Joint Estimation

Figure 7 shows the initial skeleton posture generated from the front Kinect sensor with a standing T-pose (that is, both arms spread wide). The Kinect sensor (v2) can retrieve up to 25 skeleton joints; however, our system employs only 19 of them by excluding the noisy hand tips, thumbs, and toe-tip joints. Using the initial posture, the length of each joint is measured and maintained as joint length L to be used in the optimal joint estimation. The Kinect sensor does not distinguish the left and right sides of the user; thus, the initial posture captured from the front sensor becomes the reference model, which determines the sides of the joints to be retrieved from other Kinect sensors. Our posture reconstruction method is based on the requirement of distinguishing the side and noisy level in the input data. Accordingly, it estimates the optimal joint positions in three joint groups: *center*, *torso*, and *limbs*.

Given the reference model, as shown in Fig. 4, the top nodes in the skeleton structure (*SpineBase* and *Hip*) are first located to initiate the posture reconstruction. As noted earlier, these center joints are occasionally hindered by the self-occlusion problem, especially when a user makes a turning motion. For this reason, the joint estimation mainly relies on the center joint positions retrieved from the front and rear Kinect sensors. For each input posture, the three distances among the center joints are measured for a triangular shape comparison with the reference model. Based on the triangle ratios, the closest values to those in the reference model are selected and averaged for the center joints in the output posture.

The torso joints (that is, the joint chain from *SpineBase* to *Head*) have no left and right sides to distinguish. They use the mean values of the input joints within a threshold value. If J_N is

a normal vector associated with the parent joint of a current joint, the output positions are estimated by adjusting the mean values by L in the direction of J_N.

For the limb joints (the joint link from *Shoulder* to *Hand* for an arm, and from *Hip* to *Ankle* for a leg), a *k*-means clustering method is used to distinguish the side of input joint $J \in \mathbb{R}^3$. At first, the input joints are divided into left and right sides, $S = \{J^L, J^R\}$, as

$$\arg\min_{S} \sum_{i=1}^{2} \sum_{J \in S} \left\| J - \bar{J}_i \right\|^2, \tag{7}$$

where $\bar{J}_1$ and $\bar{J}_2$ are the mean of points in J^L and J^R, respectively. Once the side of the *Shoulder* joint for an arm, or the *Hip* joint for a leg, is determined from the reference model, the minimal-difference pattern in the square values of the distance between the joint positions of the parent joint in the previous posture, J_{t-1}^{L} and J_{t-1}^{R}, and the mean joint positions from the current posture, $\bar{J}_t^1$ and $\bar{J}_t^2$, are used to determine the side for the remaining joints (*Elbow*, *Wrist*, and *Hand* for an arm, and *Knee* and *Ankle* for a leg) as follows:

$$\begin{aligned} &\min(d_1, d_2), \\ d_1 &= D(\bar{J}_t^1, J_{t-1}^{\mathrm{L}})^2 + D(\bar{J}_t^2, J_{t-1}^{\mathrm{R}})^2, \\ d_2 &= D(\bar{J}_t^1, J_{t-1}^{\mathrm{R}})^2 + D(\bar{J}_t^2, J_{t-1}^{\mathrm{L}})^2, \end{aligned} \tag{8}$$

where $D(\cdot)$ measures the Euclidean distance between two joint positions. Here, the parent joints at the previous posture serve as the reference points to determine the side of the child joint at the current posture.

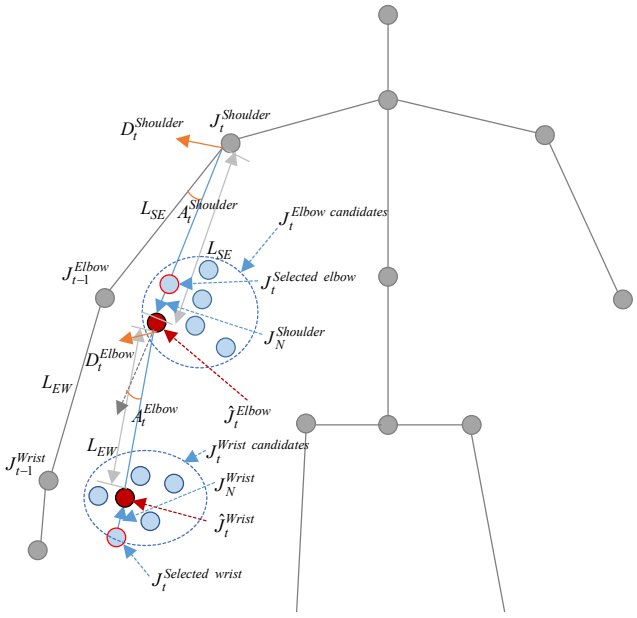


Fig. 8. Examples of joint estimation for *Elbow* and *Wrist* joints: selected joints (red circle) are adjusted to the new positions by L in the direction of J_N.

The actual positions of the limb joints are determined by selecting one of the input joint data based on

$$\hat{J}_t = \min(D_t + A_t), \tag{9}$$

where D_t and A_t are the rotation direction and the rotation angle calculated from J_{t-1} to J_t, respectively. Similar to the torso joints, the positions of $\hat{J}_t$ are adjusted by L in the direction of J_N, as shown in Fig. 8.

The Kinect sensor traces the body parts based on the learned sample data [24]; however, it often fails to position the joints at all. In such a case, the joint positions are synthesized by blending the previous postures at $t-2$ and $t-1$.

VI. Experimental Results

The applicability of our system is demonstrated by its capturing of dynamic user movements in real time. Depending on the space availability, four to eight Kinect sensors are used to capture user motion. All motion data are captured at the rate of 30 frames per second (fps). The system is best elucidated through examples of its use, as described below and presented in the accompanying video.

1. Motion Tracking

Our multi-Kinect system is compared with the commercial motion capture system to evaluate its tracking accuracy. The wearable capture system (Xsens) [25], which consists of a set of inertial sensors, and a set of eight Kinect sensors are simultaneously used for the comparison. The comparing motions are synchronized by using the frames closest to the recording times in the Kinect data. Owing to the differences in the joint structure and size between the compared skeleton models, joint vector v, which is defined from *Shoulder* to *Wrist* in an arm, or from *Hip* to *Ankle* in a leg, is used to compare the angular differences between two motions as follows,

$$Err = 1 - \frac{\sum_{0}^{N_R} (v_C \cdot v_K + 1)}{2N_R}, \tag{10}$$

where v_C and v_K are measured from the N_R frames captured from the wearable and our system, respectively.

Figure 9 compares a sequence of user motion captured by the two systems. The captured data include various dynamic movements, such as punches, kicks, steps, and turns, which are commonly performed in dance and Taekwondo. As shown in the figures, our system consistently tracks the user's dynamic movements and does not suffer from occlusion problems. It is evident that the skeleton models are different in structure and size owing to the different numbers of joints that are traceable

ETRI Journal, Volume 39, Number 2, April 2017
https://doi.org/10.4218/etrij.17.2816.0045

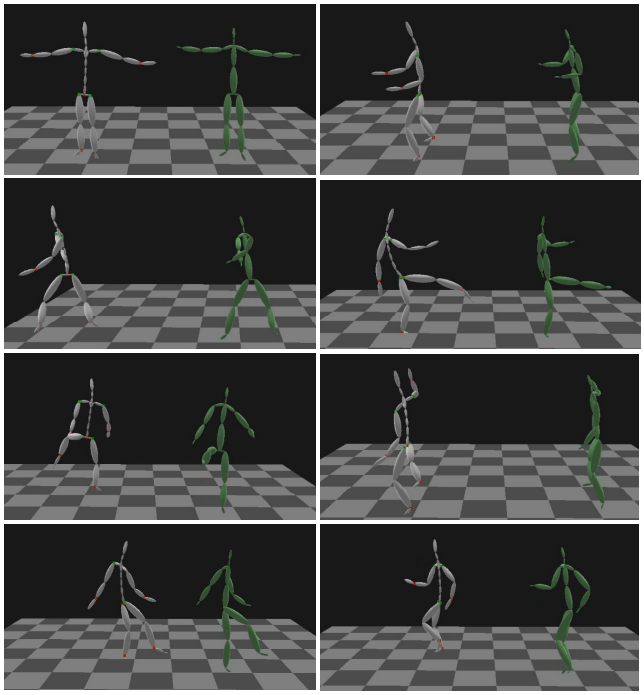


Fig. 9. Comparison of dynamic motions: commercial system (left) and our multi-Kinect system (right).

Table 1. Tracking accuracy of the single Kinect system compared to the commercial system (XSens).

Motion		Accuracy (%)				
Type	Frames	RHand	LHand	RFoot	LFoot	Average
Dance 1	2,572	64.2	66.4	58.6	57.7	61.7
Dance 2	2,072	67.3	66.8	60.2	61.3	63.9
Dance 3	4,371	70.1	69.7	54.5	56.2	62.6
Taekwondo 1	7,518	69.2	70.1	57.7	58.1	63.8
Taekwondo 2	10,060	67.2	68.3	60.1	59.2	63.7
Taekwondo 3	12,986	68.3	69.2	59.6	58.8	64.0

by the commercial system.

Tables 1 and 2 show the tracking accuracy of the single Kinect system and our multi-Kinect system against the commercial one, respectively. In this test, six sample motions (a set of K-pop dances and Taekwondo movements) with a total of 39,579 frames (approximately 22 min) are used for the comparison. As shown in the tables, our system achieved approximately 85.3% of posture similarity compared to the commercial system, thereby providing a more than 20% improvement over the single Kinect system. Once a rigid transformation is preprocessed from the ICP method, it requires less than 5 ms to reconstruct an output posture, generating the motion capture data in real time.

It is noticeable that the lower joints are less accurate than the upper ones. This is mainly because larger noises arise in the sensors attached to the feet in the wearable system.

Table 2. Tracking accuracy of our multi-Kinect system compared to the commercial system (XSens).

Motion		Accuracy (%)				
Type	Frames	RHand	LHand	RFoot	LFoot	Average
Dance 1	2,572	90.2	91.7	81.2	83.1	86.6
Dance 2	2,072	90.3	92.5	79.2	82.0	86.0
Dance 3	4,371	91.0	91.6	78.2	78.1	84.7
Taekwondo 1	7,518	91.1	91.9	79.2	79.2	85.4
Taekwondo 2	10,060	89.0	90.0	79.0	78.9	84.2
Taekwondo 3	12,986	90.0	90.9	79.6	80.0	85.1

Furthermore, the feet sensors in the wearable system are placed relatively high to increase the recognition of kicks in our system.

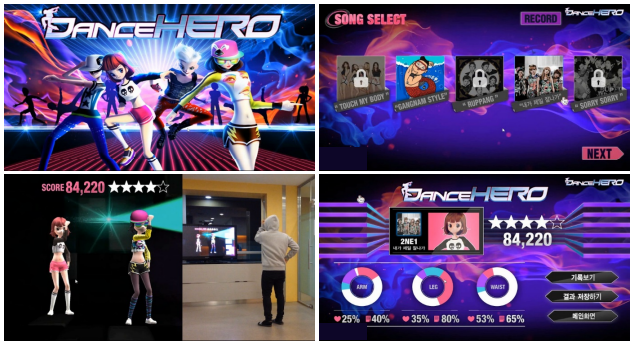
2. Motion-Based Contents

To demonstrate the applicability of our multi-Kinect system, we integrate the system into two motion-based contents: dance and Taekwondo programs. In these contents, a user either trains in dance or Taekwondo movements by imitating an expert's motions, which are captured by the commercial system and archived in advance as a database. As shown in Fig. 10, as the user is instructed to follow expert motion, the training system compares the joint movements to the expert's ones and identifies the similarity level as a percentage at the end of each training session. Throughout the sessions, the user's movements are compared based on the five feature vectors, two for each limb and one for torso, extracted from the skeleton posture [22]. For the motion-based contents, a set of four Kinect sensors, one for each side, are installed and synchronized to capture user motion.

Figure 11 shows the effectiveness of our training system. A total of 12 students, not formerly trained in dance or Taekwondo, participated in the training program for a week. For each day, each student was given five to ten training sessions, and the best similarity level was recorded. The performance records showed that the participants improved their movements in the given dances and Taekwondo up to 22% in the similarity level at the end of the week.

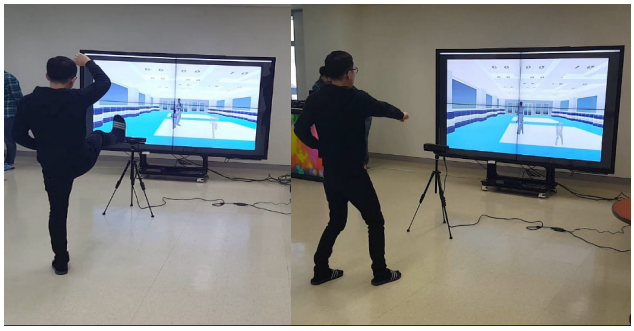
VII. Conclusion

Human motion has been rigorously analyzed and studied by many researchers. The emergence of motion capture technology has enabled the efficient digitizing of every detail of a performer's movements. Nevertheless, commercial systems are not readily available to a general user owing to the high cost and requirement of a specially designed suit to wear with many markers. An off-the-shelf solution, such as Kinect, offers a



(a)

(b)

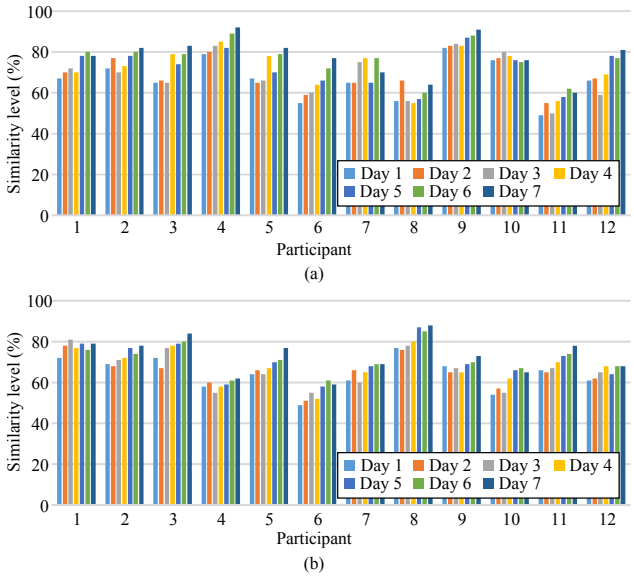


(c)

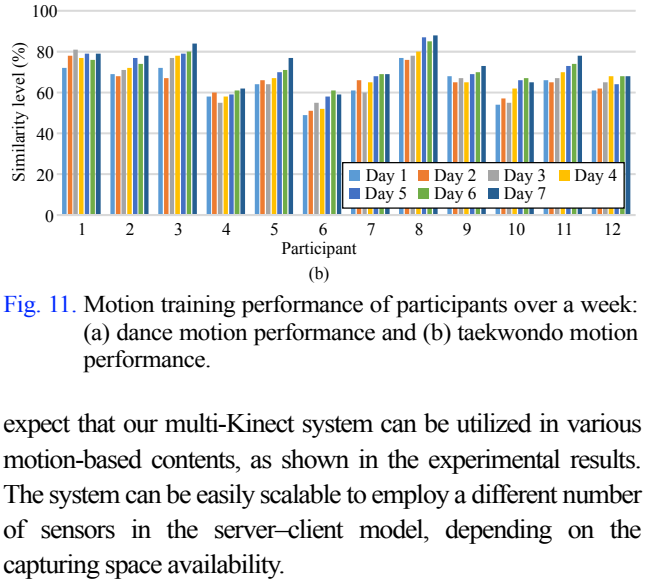
Fig. 10. Motion-based contents with our multi-Kinects system: (a) K-pop dance, (b) traditional korean dance, and (c) taekwondo.

marker-free and inexpensive way to capture user motion. However, use of a single of sensor suffers from occlusion and body rotation problems. Employing multiple sensors can overcome aspects of these problems; however, many existing approaches mainly target relatively simple and slow motion.

In this paper, we proposed a multi-Kinect system that can track fast and dynamic movements of a general user in real time. By adopting multiple Kinect sensors, our calibration method represents a single skeleton model from a set of depth images, and our posture reconstruction method estimates the optimal joint positions for the noisy skeleton data. During the calibration and posture reconstruction, the system requires no user intervention and generates output motion with an accuracy comparable to that of the commercial system (XSens). We



(a)



(b)

Fig. 11. Motion training performance of participants over a week: (a) dance motion performance and (b) taekwondo motion performance.

expect that our multi-Kinect system can be utilized in various motion-based contents, as shown in the experimental results. The system can be easily scalable to employ a different number of sensors in the server–client model, depending on the capturing space availability.

The current Kinect sensor (v2) with a supported library (SDK) requires one PC for each sensor because of the high data bandwidth. This makes the overall system complicated in terms of connections and increases the system cost. Using an open library [26] can relieve the number of connecting PCs; however, it does not support directly retrieving the skeleton data from the sensors, such as the Kinect SDK. We are currently working on constructing a 3D skeleton model from an articulated template model and multiple depth images.

Furthermore, some dynamic movements, such as the jump kicks and high kicks, are not well tracked by our system, resulting in incorrect joint positions during the posture reconstruction. The Kinect sensor only supports a capturing speed of 30 fps, which is not reliable for fast user motion. Furthermore, the skeleton retrieval performance with the Kinect SDK depends on a trained set of human postures [27]. Utilizing small and weightless inertial devices, which are attached to foot parts, can complement such shortcomings by detecting the foot joint positions. We are currently working to integrate these devices into our multi-Kinect system for improved posture reconstruction.

References

[1] Vicon Motion Capture System. https://www.vicon.com

[2] OptiTrack Motion Capture System. http://optitrak.com
[3] E. de Aguiar et al., "M^3: Marker-Free Model Reconstruction and Motion Tracking from 3D Voxel Data," *Conf. Comput. Graph. Applicat.*, Los Alamitos, CA, USA, Oct. 6–8, 2004, pp. 101–110.
[4] B. Michoud et al., "Real-Time Marker-Free Motion Capture from Multiple Cameras," *Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 14–21, 2007, pp. 1–7.
[5] S. Corzaaz et al., "A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach," *Ann. Biomed. Eng.*, vol. 34, no. 6, June 2006, pp. 1019–1029.
[6] E.D. Aguiar et al., "Marker-Less Deformable Mesh Tracking for Human Shape and Motion Capture," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Minneapolis, MN, USA, June 17–22, 2007, pp. 1–8.
[7] J. Gall et al., "Motion Capture Using Joint Skeleton Tracking and Surface Estimation," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Miami, FL, USA, June 20–25, 2009, pp. 1746–1753.
[8] Microsoft Kinect Camera. https://developer.microsoft.com/en-us/windows/kinect
[9] L. Chen, H. Wei, and J. Ferryman, "A Survey of Human Motion Analysis Using Depth Imagery," *Pattern Recogn. Lett.*, vol. 23, no. 15, Nov. 2013, pp. 1995–2006.
[10] D.S. Alexiadis et al., "Evaluating a Dancer's Performance Using Kinect-Based Skeleton Tracking," *Proc. ACM. Int. Conf. Multimedia*, Scottsdale, AZ, USA, Nov. 28–Dec. 1, 2011, pp. 659–662.
[11] S. Izadi et al., "KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera," *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, Santa Barbara, CA, USA, Oct. 16–19, 2011, pp. 559–568.
[12] E. Auvinet, J. Meunier, and F. Multon, "Multiple Depth Cameras Calibration and Body Volume Reconstruction for Gait Analysis," *Int Conf. Inform. Sci., Signal Process. Their Applicat.*, Montreal, Canada, July 2–5, 2012, pp. 478–483.
[13] K. Berger et al., *Markerless Motion Capture Using Multiple Color-Depth Sensors*, *Vision, Modeling, and Visualization*, Aire-la Ville, Switzerland: The Eurographics Association, 2011, pp. 317–324.
[14] L. Zhang et al., "Real-Time Human Motion Tracking Using Multiple Depth Cameras," *IEEE Int. Conf. Intell. Robots Syst.*, Vilamoura-Algarve, Portugal, Oct. 7–12, 2012, pp. 2389–2395.
[15] B. Williamson et al., *Multi-kinect Tracking for Dismounted Soldier Training*, *Interservice/Ind. Training, Simulation, Educ. Conf.*, Orlando, FL, USA, Dec. 3–6, 2012, pp. 1–9.
[16] S. Asteriadis et al., "Estimating Human Motion from Multiple Kinect Sensors," *Proc. Int. Conf. Comput. Vis. Comput. Graph.s Collaboration Techn. Applicat.*, Berlin, Germany, June 6–7, 2013, pp. 3–8.
[17] A. Kitsikidis et al., "Dance Analysis Using Multiple Kinect Sensors," *Int. Conf. Comput. Vis. Theory Applicat.*, Lisbon, Portugal, Jan. 5–8, 2014, pp. 789–795.
[18] S. Kaenchan et al., "Automatic Multiple Kinect Cameras Setting for Simple Walking Posture Analysis," *Int. Comput. Sci. Eng. Conf.*, Bangkok, Thailand, Sept. 4–6, 2013, pp. 245–249.
[19] S. Moon et al., "Multiple Kinect Sensor Fusion for Human Skeleton Tracking Using Kalman Filtering," *Int. J. Adv. Robot. Syst.*, vol. 13, 2016, pp. 1–10.
[20] H. Jo et al., "Motion Tracking System for Multi-user with Multiple Kinects," *Int. J. u- e-Service, Sci. Technol.*, vol. 8, no. 7, 2015, pp. 99–108.
[21] N. Ahmed, *Unified Skeletal Animation Reconstruction with Multiple Kinects*, Aire-la Ville, Switzerland: the Eurographics Association, 2014, pp. 5–8.
[22] S. Baek and M. Kim, "Dance Experience System Using Multiple Kinects," *Int. J. Future Comput. Commun.*, vol. 4, no. 1, Feb. 2015, pp. 45–49.
[23] P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," *Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, Feb. 1992, pp. 239–256.
[24] W.H. Press et al., *Numerical Recipes in C++: the Art of Scientific Computing*, New York, USA: Cambridge University Press, 2002.
[25] Xsens MVN Motion Capture System. http://xsens.com
[26] Libgreenect2. https://openkinect.github.io/libfreenect2/
[27] J. Shotton et al., "Real-Time Human Pose Estimation in Parts from Single Depth Images," *Commun. ACM*, vol. 56, no. 1, Jan. 2013, pp. 116–124.

Yejin Kim received his BS degree in computer engineering from the University of Michigan, Ann Arbor, USA, in 2000. He received his MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2003 and the University of California, Davis, USA, in 2013, respectively. From 2003 to 2013, he worked at the Visual Contents Research Department, ETRI, Daejeon, Rep. of Korea, as a research scientist. Currently, he is an assistant professor at Hongik University, Sejong, Rep. of Korea. His research interests include 3D character animation and authoring techniques in computer graphics.

Seongmin Baek received his MS degree in computer science (virtual reality) from Pohang University of Science and Technology, Rep. of Korea in 2001. Currently, he is working as a principal member of the engineering staff at the Visual Contents Research Department, ETRI, Daejeon, Rep. of Korea. His research interests include digital contents, animation, and physics-based simulation.

Byung-Chull Bae received his BS and MS degrees in 1993 and 1998, respectively, in electronics engineering from Korea University, Seoul, Rep. of Korea. He received his PhD degree in computer science from North Carolina State University, Raleigh, NC, USA in 2009. He worked at LG Electronics, Seoul, Rep. of Korea and Samsung Electronics, Suwon, Rep. of Korea as a research engineer. He is currently an assistant professor at Hongik University, Sejong, Rep. of Korea. His research interests include artificial intelligence in games and storytelling.