International Journal of Engineering & Technology, 7 (3.34) (2018) 521-526

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Interactive Dance Guidance Using Example Motions

[*1]Yejin Kim

*[*1]School of Games, Hongik University, 2639 Sejong-ro, Jochiwon-eup, Sejong, 30016, Republic of Korea*
**Corresponding author E-mail: yejkim@hongik.ac.kr*

Abstract

Background/Objectives: Human movements in dance are difficult to train without taking an actual class. In this paper, an interactive system of dance guidance is proposed to teach dance motions using examples.
Methods/Statistical analysis: In the proposed system, a set of example motions are captured from experts through a method of marker-free motion capture, which consists of multiple Kinect cameras. The captured motions are calibrated and optimally reconstructed into a motion database. For the efficient exchange of motion data between a student and an instructor, a posture-based motion search and multi-mode views are provided for online lessons.
Findings: To capture accurate example motions, the proposed system solves the joint occlusion problem by using multiple Kinect cameras. An iterative closest point (ICP) method is used to unify the multiple camera data into the same coordinate system, which generates an output motion in real time. Comparing to a commercial system, our system can capture various dance motions over an average of 85% accuracy, as shown in the experimental results. Using the touch screen devices, a student can browse a desired motion from the database to start a dance practice and send own motion to an instructor for feedback. By conducting online dance lessons such as ballet, K-pop, and traditional Korean, our experimental results show that the participating students can train their dance skills over a given period.
Improvements/Applications: Our system is applicable to any student who wants to learn dance motions without taking an actual class andto receive online feedback from a distant instructor.

Keywords: *Dance motion, dance train, example motion, interactive guidance, online lesson.*

1. Introduction

Culture contents such as dance, play, and song are attracting much attention worldwide as seen from the recent increase in popularity of Korean culture (i.e. Hallyu). According to UNESCO, culture contents become oblivious if not properly recorded and passed down to descendants [1]. Over years, learning such culture contents has been mainly relied on a simple view on images and videos retrieved from online web sites (i.e. Youtube and Google Video). In a such unidirectional learning process, a student lacks the expert's feedback and often misses key factors in the contents. This causes serious learning deficiencies, especially for the active contents such as dance, where a student should follow complicated movements shown from an instructor. Up to this day, taking an actual class has been the most effective way to train a sequence of complicated human movements under the physical guidance of an instructor. However, it is not easy to attend an actual class due to the time limit or to find a proper instructor to receive feedback for guidance.

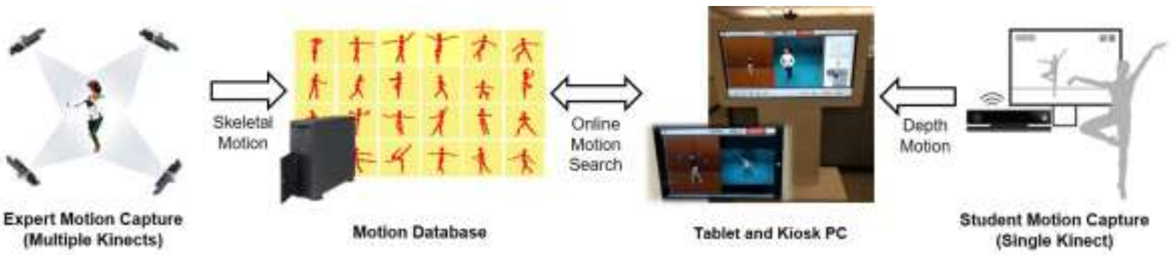

Figure 1: Overall system for interactive dance guidance using example motions

In this paper, an interactive system of dance guidance is proposed to train dance motions using example motions. As seen in Figure 1, a set of example motions are captured from expert dancers by using a method of marker-free motion capture, which adopts multiple depth cameras to capture various dance movements in real time. During the capturing process, the multiple camera data, which are captured at different viewpoints, are unified into the same coordinate system. From this space, an optimal skeleton posture is reconstructed by removing the noisy joint data. The captured motions are archived into a motion database server for online lessons. Given the touch screen devices, a student can browse a desired dance motion from the database to start a dance practice and send own motion to an instructor for online feedback. In this training process, a posture-based motion search and multi-mode views are provided for the efficient exchange of motion data under a networked environment. As shown in the experimental results, our system can capture human motions with a comparable accuracy against a commercial system and improve a student's dance skills over a given period.

Training dance motions without a physical attendance of an instructor has been suggested in different fields. In the classical dance fields, movement notes (i.e. Labanotation and EWMN) are popularly used to express a sequence of body movements over time in the classical dance field [2, 3]. Later, these notations are digitalized into motion editing applications such as Isadora and DanceForms [4, 5]. However, these notes and applications require an extensive knowledge in human motion properties to edit human movements, which is not applicable to general users. Based on the real-time motion capture technique, several video games such as Dance Central and Just Dance series, are introduced to imitate popular dances [6, 7]. In these systems, a single depth camera such as MS Kinect, is used to capture a player's motion [8]. During the game play, the captured motion is compared against the expert's motion stored in the game content for scores. However, these systems are mainly designed for an entertainment purpose and do not provide a precise guidance on the user's motions. Recently, a virtual reality is added to the dance training system with the motion capture technique [9]. In this system, a student imitates an instructor's motion projected on the wall. However, the student should wear a special suit with a number of markers attached in order to imitate the expert motion. In addition, the overall system cost is expensive and requires a large space due to the optical motion capture system used; thus, it is not suitable for online lessons that are quick and easy to start.

To capture human motions, both optical and magnetic systems are widely used to produce high-quality motion data. However, they require a dancer to wear a specially designed suit attached with a set of markers or sensors, possibly restricting a range of movements and affecting the performance. On the other hand, the marker-free motion capture provides a better freedom of movements without the suit to wear, more suitable to capture dynamic movements in dance. Recently, an availability of inexpensive depth cameras makes it easier to capture dance motion [8, 10]. For these reasons, many approaches adopt multiple depth cameras to capture human motion [11-17]. In [11], a particle filtering and partition sampling approach is used to track human postures; however, this type of approach requires a template model to generate output postures. Using multiple Kinects, a method with hidden conditional random fields (HCRF) is used to recognize dance patterns [12]. In [13], walking motions are analyzed by tracking the mean joint positions. However, these approaches used relatively slow motions as input data. To increase an accuracy in Kinect data, the Kalman filtering is used for simple motion types [14]. In [15], multiple players are tracked from multiple Kinect cameras; however, their main purpose is to track the user positions, not their motions. To capture fast-moving movements in dance, multiple color and depth cameras are connected and synchronized for a better accuracy [16-17]. Due to the number of cameras used, their systems are expensive and complicated in setup. On the other hand, our system is designed to capture dance motions using a relatively small number of cameras for a ubiquitous use.

This paper is organized as follows. The example motion acquisition is described in Section 2. The details of interactive guidance for online dance lessons are explained in Section 3. The experimental results are shown in Section 4, and we conclude the paper with a discussion of potential improvements in Section 5.

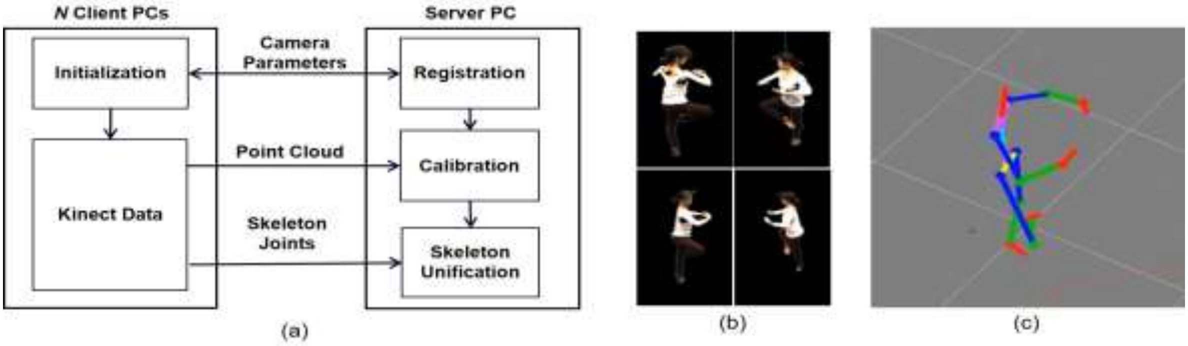

Figure 2: Example motion acquisition: (a) a server-client model for Kinect data transmission, (b) point clouds with color data received from multiple Kinects, and (c) a unified skeletal model

2. Example Motion Acquisition

2.1. Multi-Kinects Registration and Calibration

As shown in Figure 1, two Kinect cameras were placed on front and rear sides to capture dance motions all around. Since each Kinect camera is required to connect to a single PC, our system adopts a server-client model to receive the motion data from each camera as shown in Figure 2. When each of the camera PCs is connected to the server via the Ethernet connection, it registers the camera parameters (i.e. principal points, focal lengths, skew coefficients, and distortions) and identification with the server. Once the server acknowledges the connection, the camera sends a stream of color, depth, and skeletal motion data to the server. Before these data transmission, the camera PC removes the background objects and noises from the naïve depth data and generates a point cloud from it to decrease the overall data size. The point cloud is sampled from a foreground of depth image, where a stored background (i.e. a scene without a dancer) is subtracted from a scene with a dancer using a threshold value [16]. A window size of 8 pixels around the edges of a dancer are used to smooth out the noises at the edge pixels. In addition, the ankle joint positions from the Kinect skeleton are used to exclude the noise around the foot area. The ground normal is estimated from the joints between the root and spine joints of the Kinect skeleton if a dancer stands upright in the initial capture. This vector is used to correct the tilted camera orientation on the ground. Using a sampling cube, which averages 3D positions of all depth points in the cube, the foregoing process yields 1,500 to 15,000 points with the background objects and noises eliminated. For the skeletal motion, only 19 joints, excluding hand tips, thumbs, and toe tips from 25 Kinect skeleton joints, are transmitted to the server as these hand and foot joints are noisy and not important in analyzing overall dance movements.

2.2. Skeleton Unification

As the Kinect cameras are placed apart, the coordinate system of each Kinect data is different from each other. To unify the skeletal motion data from the cameras, one of the cameras is selected as a reference coordinate system, and all other camera data are transformed into this coordinate. For this unification process, a transform matrix is estimated between two point clouds (i.e. reference and each camera) by using the iterative closest point (ICP) method [18]. It is noticeable that the point clouds are used as input data to ICP instead of joint positions as the performance of ICP depends on the number of matching points used during the iterative process. To facilitate this unification process, a thin wind (i.e. 50cm) with a sphere object attached to its tip is used as a calibration tool. Due to the low resolution of Kinect depth sensor (i.e. 512 by 424 pixels), only the sphere part of the wind is recognized by the camera. A sequence of mean values from the depth data (i.e. 300 to 500 points) by tracking the sphere object is used to align two point clouds as the preprocessing step, which accelerates the ICP process with dense points from the point clouds. Provided with the transformation matrix, the skeletal motion data (i.e. joint positions) from each camera are transformed into the reference coordinate by applying the transformation matrix.

2.3. Posture Reconstruction

Our system reconstructs a motion posture based on a dancer's initial posture and the optimal joint selection from the unified skeletal data in the same coordinate system. Figure 3 shows the initial posture used, where a dancer is facing one of the cameras with a T-pose. From this posture, the length of each joint is measured by locating the joint positions of Kinect skeleton and set as the reference model. The left and right sides of the joints are decided from this model during the reconstruction process. For example, the right joint tracked from the front camera can be tracked as the left joint from the rear camera since the camera does not distinguish the body from front or back sides.

To reconstruct the skeletal joints, the top nodes (i.e. root and hip joints) in the hierarchical structure are first located and the remaining joints are determined from the multiple candidate joints received from the cameras. However, it is not easy to find the accurate positions of the joints due to the noises incurred by the self-occlusion problem. For example, as shown in Figure 3, when the user rotates or crosses a body part, the camera starts to track wrong joint positions and reconstruct them into an incorrect posture. This continues until there is no self-occlusion in the user posture, which causes visual artifacts such as jerkiness and unnaturalness in the output motion.

For the root and hip joints, two distances, one between the root and hips and another between the right and left hips, should be measured from the initial model. Here, the closest positions to the triangle formed by the left hip, root, and right hip joints in the initial model, are compared and selected for the root and hip joints from the unified motion data. The ratios of joint lengths are used with a threshold value to select the optimal root and hip joints. To reduce a false selection due to the tracking noises, two sets of best triangular joints are selected and combined based on weights to estimate the joint positions.

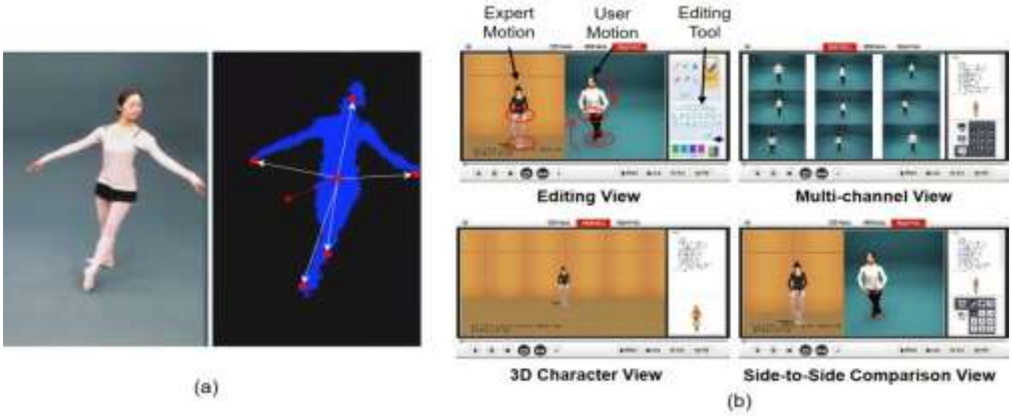

Figure 4: Interactive dance guidance: (a) a set of feature vectors used for online motion search and (b) multi-mode views for online dance lessons

For the torso joints, average positions of the unified data are used to estimate the joint positions as there is no separation between the right and left sides. Each joint in the torso part are aligned from the normal vectors associated with parent joints and adjusted by the joint lengths measured from the initial model.

For the arm and leg joints, our system adopts the *K*-means algorithm, which separates the top joint (i.e. the shoulder joint for an arm and the hip joint for a leg) into the left and right sides based on the distance of each joint from the mean of the other joint data. Once the side of these top joints are decided, the side for the remaining joints (i.e. the elbow and wrist joints for an arm and the knee and ankle joints for a leg) is decided as follows, $\min(d_1, d_2)$,

$$d_1 = D\left(\bar{J}_f^1, J_{f-1}^L\right)^2 + D\left(\bar{J}_f^2, J_{f-1}^R\right)^2, \qquad (1)$$

$$d_2 = D\left(\bar{J}_f^1, J_{f-1}^R\right)^2 + D\left(\bar{J}_f^2, J_{f-1}^L\right)^2,$$

Where $D(\cdot)$ measures the Euclidean distance between two joint positions. Here, J_{f-1}^L and $J_{f-1}^R$are the left and right joint positions of the parent joint at the previous frame, respectively, where $\bar{J}_f^1$ and$\bar{J}_f^2$ are the mean joint positions at the current frame. It is noteworthy that $J_{f-1}^{L,R}$is used as a reference point to decide the side of its child joint at the current frame. The actual positions of the arm and leg joints are determined by selecting one of the input joint data based on

$$J_f = \min(D_t + A_t), \qquad (2)$$

where$D_t$and$A_t$are the rotation direction and the rotation angle estimated from $J_{f-1}^{L,R}$ to $J_f^{L,R}$, respectively. As seen in the torso joints, the positions of$J_f^{L,R}$are adjusted bythe length in the direction of a normal vector of$J_f^{L,R}$. Whenall cameras fail to provide the joint data, the averages of joint positions reconstructed in the previous postures (i.e. 5 to 10 frames) are used for the current posture reconstruction.

3. Interactive Dance Guidance

3.1. User Motion Abstraction and Comparison

Given the touch screen devices, a student can browse and retrieve a desired motion from the motion database. Using an attached depth camera on the device, the student can capture one's own motion and send it to an instructor for online feedback. While both example and user motions are archived into the database, its size grows fast, making the search process a time-consuming task. For this reason, a user motion is abstracted by estimating a set of feature vectors. As seen in Figure 4, from the point cloud data, a set of extreme points,e_i, where i ∈[*Center, Head, RightHand, LeftHand, RightAnkle, LeftAnkle*], are detected from the human body parts. For an efficient search fore_i, a quadtree structure can be applied to limit the search area, while histogram of oriented gradients (HOG) and support vector machine (SVM) are used to specify $e_i$from the segmented body parts [19].

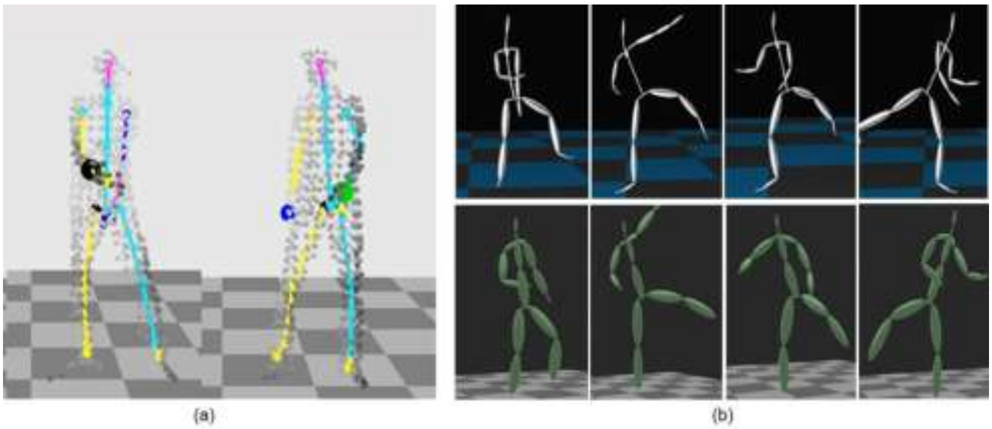

Figure 5: Motion acquisition results: (a) a reconstructed posture (arm joints occluded by a body) and (b) comparison of dynamic motions between the Xsens (top) and our system (bottom)

Givene_i, a set of feature vectors, $\boldsymbol{v}_j$, where$j$ ∈[*Head, RightHand, LeftHand, RightAnkle, LeftAnkle*], can be estimated between the two extreme points. In addition, a body orientation vector,$\boldsymbol{n}$, can be estimated ate_{Center}. To compare two postures between one from a frame in an example motion,F_A, and the other from a frame in a user motion, F_B, their similarity can be measured as a minimum sum of weighted angular differences as follows,

$$D(F_A, F_B) = \min(D(\boldsymbol{v}_A, \boldsymbol{v}_B), D(\boldsymbol{n}_A, \boldsymbol{n}_B)), \tag{3}$$

where

$$D(\boldsymbol{v}_A, \boldsymbol{v}_B) = \sum_{j=1}^{5} \omega_j \left\|\boldsymbol{v}_{A,j} - T_{\theta,\Delta x,\Delta z}\boldsymbol{v}_{B,j}\right\|^2, \tag{4}$$

and

$$D(\boldsymbol{n}_A, \boldsymbol{n}_B) = \omega_{Center}\left\|\boldsymbol{n}_{A,Center} - \boldsymbol{n}_{B,Ceneter}\right\|^2. \tag{5}$$

Here, $\omega_j$is a weight value that sets the importance of $\boldsymbol{v}_j$during the search. In addition, $T_{\theta,\Delta x,\Delta z}$ is used to align $\boldsymbol{v}_{B,j}$to $\boldsymbol{v}_{A,j}$, which rotates $\boldsymbol{v}_{B,j}$about the vertical y-axis and translates $\Delta x = x_{A,j} - x_{B,j}$ and$\Delta z = z_{A,j} - z_{B,j}$for a more precise comparison. Given a single posture captured from the user, our system lists similar postures based on the similarity order. To avoid abundant search results, a window size of $N_f$frames are used to skip the similar postures in neighbor frames.

3.2. Online Dance Lessons

In our system, the student and instructor exchange the motion data (i.e. color images and 3D skeletal motion) through the motion database server which is accessed by the touch screen devices (i.e. tablet, kiosk, and smart phone). As seen in Figure 4, the student can browse the database and selects a desired motion to start an online lesson. To facilitate this training process, our system provides various view modes: the multi-channel view for displaying an example motion at different angles, the 3D character view for displaying a motion in an articulated model, and the side-to-side comparison view for displaying two motions (i.e. example and user) at the same time. Using the attached depth camera on the devices, the student can capture and send own motion to an instructor for online feedback. Using another device, the instructor can review the student motion and send the commented motion back to the student through the database server. As shown in Figure 4, an editing view is provided to add correction marks on the frames in a motion sequence. It is noticeable that this online interaction can be made in real time if the devices are connected in broadband connections under the networked environment.

4. Experimental Results

4.1. Motion Acquisition

To build the example motion database, the proposed system adopts multiple Kinect cameras that capture expert motions by minimizing the self-joint occlusion problem. As shown in Figure 5, a dancer's posture can be reconstructed although the hand joints are occluded or when the user turns around. To evaluate the tracking accuracy, our system is compared against the commercial system, Xsens[20], which uses a wearable suit attached with a set of inertial sensors. This way, two systems can be used at the same time for capturing motions from a dancer. The capturing speed (i.e. 120 fps) of Xsens system is downscaled to the speed (i.e. 30 fps) of Kinect camera. Due to the differences in the joint structure and size used between the two systems, a joint vector$\boldsymbol{j}$, which is defined from the arm and leg joints is used to compare the angular differences between two skeletal motions as follows,

$$Err = 1 - \frac{\sum_0^{N_c}(\boldsymbol{j}_K \cdot \boldsymbol{j}_X + 1)}{2N_c}, \tag{6}$$

Where$\boldsymbol{j}_K$and$\boldsymbol{j}_X$are estimated from our Kinect system and Xsens with a total of N_c frames, respectively. Table 1compares the tracking accuracy between the two systems. For this test, six example motions are captured from ballet, traditional Korean, and K-pop motions with a total of 60,178 frames. As shown in the table, the overall average accuracy of our system is measured over 85% comparing to the commercial system. Once the skeleton unification is calibrated from the ICP process, the posture reconstruction takes less than 5ms to generate an output posture for each frame, which is suitable for a real-time motion capture. It is noticeable that the accuracy of foot joints is relatively lower than the hand ones. It is because there are larger noises around the foot sensors in the commercial system. In addition, the foot sensors are attached to a higher position than the normal one to track the kicking postures better with our Kinect system.

Table 1: The accuracy measures between the commercial and our systems

Dance Type (Frames)	RHand (%)	LHand (%)	RFoot (%)	LFoot (%)	Average (%)
Ballet 1 (3,102)	91.2	91.7	83.2	84.1	87.6
Ballet 2 (3,573)	92.3	92.5	85.2	84.0	88.5
Korean 1 (8,321)	89.1	90.6	82.2	83.1	86.3
Korean 2 (5.678)	88.1	90.9	80.2	79.8	84.8
K-pop 1(17,162)	89.0	88.2	80.6	82.3	85.0
K-pop 2 (22,342)	87.0	86.9	78.8	79.1	83.0
Average	89.5	90.1	81.7	82.1	85.8

4.2. Online Dance Lessons

For the motion database, various example motions were captured from ballet, K-pop, and traditional Korean experts. Table 2 shows different types and the total frames (seconds) captured for each dance category. To demonstrate the effectiveness of our system, a total of 10 students, who had not formerly taken any dance lesson, participated in online dance lessons such as ballet, K-pop, and traditional Korean, using our interactive system. Within four weeks, each student has taken a training session 3 to 4 times a week for each dance category and received online feedbacks from an instructor. At the end of each week, the instructor scores the performance level for each student by counting the number of correct key postures performed by the students. Figure 6 shows that most of the participants have improved their dance performance levels over the given period. It is noteworthy that the increase in their performance levels differs from one dance category to another. For example, most participants have felt the K-pop dance (i.e. most dynamic) is the most difficult to train while the traditional Korean dance is least difficult from the online lessons.

Table 2: Dance motion database: all examples archived at 30 fps

Category	Types	Total Frames (s)
Ballet	25	255,210 (8507)
K-pop	10	63,660 (2122)
Korean	10	63,440 (2111)

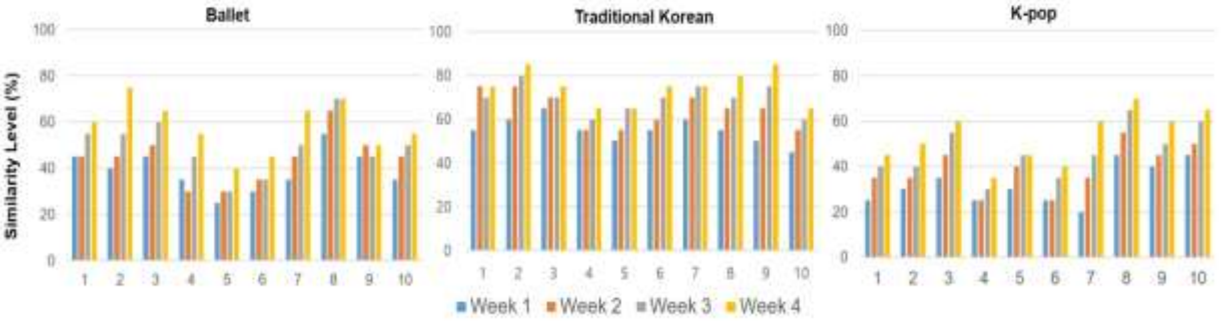

Figure 6: Online dance training performance of 10 participants over 4 weeks.

5. Conclusion

The proposed system is developed to provide online dance lessons for a student who is unable to attend an actual class. To build the motion database, a set of example motions are captured from various expert dancers via the method of marker-free motion capture. Using the touch screen devices, the student can browse and practice a desired motion from the database while an instructor can review the student motions and send the feedback back to the student under a networked environment. During this interactive training process, our system provides the posture-based motion search and multi-view modes to provide the efficient exchange of motion data between the student and instructor. As shown in the experimental results, our system can capture human motions with a comparable accuracy against a commercial system and improve a student's dance skills over a given period.

The current version of depth camera (i.e. Kinect v2) with a supported software library requires one PC for each camera due to the high bandwidth of data transmission. We are currently working on connecting multiple Kinects to one PC to reduce the overall system connections and cost. Furthermore, our system reconstructs incorrect postures from some of challenging movements such as jump kicks and turning kicks. This is mainly because our reconstruction depends on the Kinect skeletal motion, which are not trained with such inputs and generate noisy outputs. Using a small and weightless inertial sensor attached to each foot can improve the tracking accuracy; however, it can affect the freedom of dancing performance.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1C1B5017000)

References

[1] United Nations Educational, Scientific and Cultural Organization (UNESCO). Intangible cultural heritage. Retrieved from https://ich.unesco.org/en/home/.

[2] Dance Notation Bureau. Labanotation. Retrieved from http://www.dancenotation.org/.

[3] The Noa Eshkol Foundation for Movement Notation. Eshkol-Wachman Movement Notation (EWMN). Retrieved from http://noaeshkol.org/.

[4] TroikaTronicx. Isadora. Retrieved from https://troikatronix.com/.

[5] Credo Interactive. DanceForms 2. Retrieved from http://charactermotion.com/products/danceforms/.

[6] Harmonix. Dance Central Spotlight. Retrieved from http://www.harmonixmusic.com/games/dance-central/.

[7] Ubisoft Entertainment. Just Dance Now. Retrieved from https://just-dance.ubisoft.com/en-us/home/.

[8] Microsoft. Kinect Camera Sensor. Retrieved from https://developer.microsoft.com/en-us/windows/kinect/.

[9] Chan, J., Leung, H., Tang, J., & Komura, T. (2011). A Virtual Reality Dance Training System Using Motion Capture Technology. IEEE Transactions on Learning Technologies,4(2), 187-195.

[10] Intel. RealSense. Retrieved from https://software.intel.com/en-us/realsense.

[11] Zhang, L., Sturm, J., Cremers, D., &Lee, D. (2012, October 7-12). Real-time human motion tracking using multiple depth cameras. Paper presented at the IEEE International Conference on Intelligent Robots and Systems. doi: 10.1109/IROS.2012.6385968

[12] Kitsikidis, A., Dimitropoulos, K., Douka, S., &Grammalidis, N. (2014, January 5-8). Dance Analysis using Multiple Kinect Sensors. Paper presented at the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal. Piscataway, New Jersey: IEEE.

[13] Kaenchan, S., Mongkolnam, P., Watanapa, B., &Sathienpong, S. (2013, September 4-6). Automatic Multiple Kinect Cameras Setting for Simple Walking Posture Analysis. Paper presented at the International Computer Science and Engineering Conference. doi: 10.1109/ICSEC.2013.6694787

[14] Moon, S., Park, Y., Ko, D.W., &Suh, I.H. (2016). Multiple Kinect Sensor Fusion for Human Skeleton Tracking using Kalman

Filtering. International Journal of Advanced Robotic Systems, 13(2), 1-10, doi:10.5772/62415

[15] Jo, H., Yu, H., Kim, K., &Jung, H.S. (2015). Motion Tracking System for Multi-User with Multiple Kinects. International Journal of u- and e-Service, Science and Technology, 8(7), 99-108. doi:10.14257/ijunesst. 2015.8.7.10

[16] Kim, Y., Baek, S., & Bae, B.-C. (2017). Motion Capture of the Human Body. ETRI Journal, 39(2), 181-190. doi:10.4218/etrij.17.2816.0045

[17] Kim, Y. (2017). Dance motion capture and composition using multiple RGB and depth sensors, International Journal of Distributed Sensor Networks, 13(2), 1-11. doi:10.1177/1550147717696083

[18] Besl, P.J., & McKay, N.D. (1992). A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

[19] Hong, S., & Kim, M., (2016). A Framework for Human Body Parts Detection in RGB-D Image. Journal of Korea Multimedia Society, 19(12), 1927-1935.

[20] Xsens.MVN motion capture system. Retrieved from http://xsens.com.