An Approach to Ballet Dance Training through MS Kinect and Visualization in a CAVE Virtual Reality Environment

MATTHEW KYAN, Ryerson University
GUOYU SUN and HAIYAN LI, Communication University of China
LING ZHONG, Guangdong University of Technology
PAISARN MUNEESAWANG, Naresuan University
NAN DONG, BRUCE ELDER, and LING GUAN, Ryerson University

This article proposes a novel framework for the real-time capture, assessment, and visualization of ballet dance movements as performed by a student in an instructional, virtual reality (VR) setting. The acquisition of human movement data is facilitated by skeletal joint tracking captured using the popular Microsoft (MS) Kinect camera system, while instruction and performance evaluation are provided in the form of 3D visualizations and feedback through a CAVE virtual environment, in which the student is fully immersed. The proposed framework is based on the unsupervised parsing of ballet dance movement into a structured *posture space* using the spherical self-organizing map (SSOM). A unique feature descriptor is proposed to more appropriately reflect the subtleties of ballet dance movements, which are represented as *gesture trajectories* through posture space on the SSOM. This recognition subsystem is used to identify the category of movement the student is attempting when prompted (by a virtual instructor) to perform a particular dance sequence. The dance sequence is then segmented and cross-referenced against a library of gestural components performed by the teacher. This facilitates alignment and score-based assessment of individual movements within the context of the dance sequence. An immersive interface enables the student to review his or her performance from a number of vantage points, each providing a unique perspective and spatial context suggestive of how the student might make improvements in training. An evaluation of the recognition and virtual feedback systems is presented.

Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems

General Terms: Design, Algorithm, Performance

Additional Key Words and Phrases: MS Kinect, ballet, dance, immersive training and simulation, virtual reality, human–computer interaction, gesture recognition, self-organizing maps, CAVE

ACM Reference Format:
Matthew Kyan, GuoYu Sun, Haiyan Li, Ling Zhong, Paisarn Muneesawang, Nan Dong, Bruce Elder, and Ling Guan. 2015. An approach to ballet dance training through MS kinect and visualization in a CAVE virtual reality environment. ACM Trans. Intell. Syst. Technol. 6, 2, Article 23 (March 2015), 37 pages.
DOI: http://dx.doi.org/10.1145/2735951

Authors' addresses: M. Kyan, Ryerson University, 350 Victoria St., Toronto, ON, Canada, M5B2K3; email: mkyan@ryerson.ca; G. Sun and H. Li, Communication University of China, 1 Dingfuzhuang East Street, Chaoyang, Beijing, P.R. China, 100024; emails: gysunlu@gmail.com, hyli@cuc.edu.cn; L. Zhong, Guangdong University of Technology, 100 Waihuan Xi Road, Guangzhou Higher Education Mega Center, Panyu, Guangzhou, P.R China, 510006; email: lzhong@rnet.ryerson.ca; P. Muneesawang (corresponding author), Naresuan University, 99 Phisanulok-nakornsawang Road, Muang, Phisanulok, Thailand, 65000; email: paisarnmu@nu.ac.th; N. Dong, B. Elder, and L. Guan, Ryerson University, 350 Victoria St., Toronto, ON, Canada, M5B2K3; emails: {ndong, belder}@ryerson.ca, lguan@ee.ryerson.ca.

DOI: http://dx.doi.org/10.1145/2735951

1. INTRODUCTION

The comparison and synthesis of captured motion data taken in real time from the trainee against the reference (trainer) data are the main issues in the virtual reality (VR) dance training system. The development of accurate analytical tools and systems is highly desirable [Ho et al. 2013; Komura et al. 2006]. This article presents a method to address these issues based on two techniques: automatic dance gesture recognition and the 3D visual feedback to effectively assess student performance and training. This is particularly applicable to the classical ballet dance.

Classical ballet techniques are based on distinct aesthetic ideals. Compared with other forms of dance, ballet is a highly technical and disciplined style of dance. Movements are of an exacting precision and rely on strong core stability and good posture. The fundamental theoretical principles of classical ballet techniques include [Ward 2012] alignment-maintaining verticality of the torso, placement-minimal displacement of the pelvis from the center position, turnout-maximum external rotation of the lower limbs, and extension-maximum elongation of the lower limbs.

In traditional teaching, the demonstration-performance method [Yang et al. 2012] is employed to teach students physical and mental skills. A demonstration must be given by the instructor, which will then be imitated by the students under close supervision. Instructors provide their students with feedback based on their performances by informing them how their movement response compares to the ideal template or prototype for their particular discipline. The efficacy of this feedback depends largely on the instructor's ability to identify the aspects of the response that prevent the learner from attaining the skill objective [Armstrong and Haffman 1979]. While the longevity and frequent use of certain instruction theories and cues may imply that they have validity as useful teaching aids, it seems clear they would also stand up to scientific scrutiny. At this point, however, ballet dance training relies more on a qualitative than a quantitative sense. As such, the application of computerized systems for assessment and training of dance remains a topic that attracts considerable research interest.

A number of research works based on quantitative methods have appeared that have attempted to develop an objective and systematic means of analysis of dance techniques [Kulig et al. 2011; Bronner and Ojofeitimi 2011; Shippen and May 2010; Gamboian et al. 2000; Simmons 2005; Bertucco 2010]. All these works investigate the biomechanical properties of human movement by quantification and analytical description of body movement patterns using kinematic as well as kinetic data. These are captured by computerized instrumentation (i.e., video and 3D motion analysis). Here, the aims are to provide a biomechanical description of dance movements to inform dancers and dance instructors of the typical way to perform a standard movement [Bronner and Ojofeitimi 2011; Wilson et al. 2007; Couillandres et al. 2008] and to explore the relationship between dance movements and injury [Kulig 2011; Mayers and Bronner 2010]. It is now possible for a computerized system to capture kinematic data from dance teachers to use as a reference and obtain relevant kinematic variables to analyze ballet techniques [Ward 2012].

The aforementioned works explore the use of quantitative measurement tools that could potentially be used to evaluate the progress and technical development of individual dancers, as well as for evaluating teaching practice (i.e., measuring the ability of the ballet teacher to accurately identify errors in the performance of typical balletic movements). The latter issue is important, since the recognition and positive reinforcement of correct technique may provide encouragement, increased motivation, and confidence to students. Under this paradigm, the value of the measurements taken is based entirely on the representational validity of the characteristics selected for the feature set (i.e., how well the set of features selected reflect the dance gesture's most aesthetically relevant dynamic properties) and the accuracy of the feature extraction. However, the virtual reality training method can be even more sophisticated than this. In particular, the feedback the training system provides doesn't need to be exclusively quantitative in the form of score but may also involve a visual comparison of virtual characters [Chan et al. 2011] or the synthesis of dance partners "on the fly" [Ho et al. 2013].

We take this idea further with computational intelligence to enable the system to recognize the student's dance gestures with a trajectory of postures over possible gestures. This computational intelligence is implemented in a fully immersive virtual reality system, the cave automatic virtual environment (CAVE). With an explicit model of a student's gestures, assuming a desired goal, the approach uses a trajectory of postures within a self-organizing spherical structure to predict the target gestures, given their actions. So the dance teaching problem is inverted into the problem of predicting the student's gestures. This is followed by an assessment of the student's performance and visual feedback in the CAVE, allowing high degrees of view and freedom of interaction. Section 3 describes how to obtain this predictive model.

To date, there has been a distinct research emphasis on the visualization phase and, therefore, finding better virtual representations of dances. So much emphasis is placed on the technique of mimicking the dance teacher that quantitative measures and feedback are crude or nonexistent, essentially requiring the students to follow the virtual teacher [Kavakli et al. 2004; Chua et al. 2003; Hachimura et al. 2004; Yang et al. 2002]. Under this paradigm, learning ability is entirely based on the virtual representation of characters driven by the student's motion capture data and the ability of the student to follow a virtual teacher. Based on this mimicking learning, however, repetition of material without feedback does not necessarily result in improved performance. In some recent papers [Chan 2011; Alexiadis et al. 2011; Naemura and Suzuki 2006; Raptis et al. 2011; Becker and Pentland 1996], an alternative to this learning paradigm was proposed, in which the student assessment can be performed with rapid feedback using a standard automated protocol. The main activities in this approach consist of analyzing a student's motion against the desired (teacher's) dance steps and synthesizing the virtual character accordingly. Students participating in this process would receive feedback on the accuracy of their performance and on specific areas for which their accuracy is poor and thus in need of attention. Given the importance of structured learning in skill acquisition [Ericsson 1993], this tool could therefore be a valuable source of feedback and, as such, a very useful resource for dance training.

The proposed system can accommodate all the important requirements that arise in connection with standard methods of teaching elemental ballet. A novel framework is proposed for the real-time assessment and visualization of ballet dance movements, as performed by a student in an instructional VR setting. We utilize MS Kinect to capture skeletal joint tracking for acquisition of human movement data. The performance evaluation is provided in the form of 3D visualizations and feedback through the CAVE. In an offline process, the movements of a teacher are represented as gesture trajectories through unsupervised posture space on the spherical self-organizing map (SSOM). Four types of templates, based on the *bag-of-words* model, are utilized for indexing the gesture trajectories. In an online process, the dance sequence of a student is segmented and cross-reference against a library of gestural components performed by the teacher. This facilitates alignment and score-based assessment of individual movements within the context of the dance sequence.

An impediment to research on virtual reality is the lack of degrees of view and freedom of interaction. In real training with human instructors, students can observe the teacher from different angles. Until recently, presentations by virtual instructors were limited to what could be seen in a two-dimensional image projected on a screen. The CAVE offers augmented possibilities, because it allows the learner to view the virtual teacher from a variety of angles and for students' eye movements to be tracked.

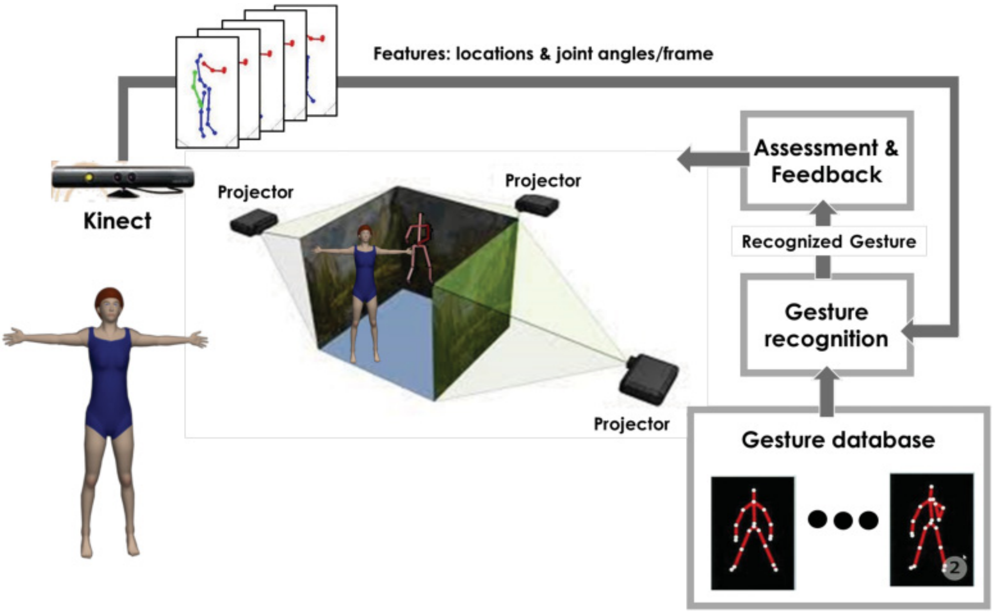


Fig. 1. System architecture.

This tracking system may be used to determine the content to be displayed on the screens, and thus the learner can perceive the virtual content. In place of the 2D visual screen and head-mounted display (HMD), the proposed system uses a CAVE to provide a better field of view and more freedom of interaction to accommodate effective feedback in dance training.

The architecture of the proposed system (shown in Figure 1) includes four components: Kinect motion capture, CAVE, gesture recognition, and gesture database. The goal of the system is to recognize a sequence of predefined movements performed within a dance sequence and to identify the occurrence and duration of these movements in the dance sequence when a beginner attempts such movements. The recognition module extracts occurrences (phases) from the beginner's performance and then assesses it against the teacher's (ground-truth) movement. Finally, the system visualizes both the teacher's and the student's dance sequences (or isolated movements) in a VR setting.

The rest of the article is organized as follows: In Section 2, we discuss related works. Section 3 describes dance representation. In Section 4, the gesture recognition method is introduced. In Section 5, visual feedback is explained. Sections 6, 7 and 8 provide the evaluation of the proposed system and the conclusion, respectively.

2. RELATED WORK

2.1. Quantitative Studies in Ballet Dance

In order to assess classical ballet movements, 2D and 3D kinematic data have been used. The 2D technique has been used in the early stages of development: video of ballet movements [Barnes 2000; Deckert 2007] and 2D computerized motion analysis [Bronner and Ojofeitimi 2006]. Recently, the majority of the quantitative studies on the kinematics of ballet have used 3D motion analysis [Kulig 2011; Wilson 2007; Golomer 2009). The types of ballet movements studied were based on the "Seven Movements of Dance" outlined in Jean-George Noverre's early manual of instruction in romantic (narrative) ballet [Noverre 1760], which are *plié*, *relevé*, *pirouettes, sauté*, *élancer*, *battement*, and *glisser*. Noverre's famous manual is the original source from which many teaching methods still used today are derived. With respect to the quantitative analysis of the *plié* movement, features are extracted to characterize the knee joint using video data [Barnes 2000] alignment of the torso [Krasnow 1997], and pelvic placement [Holt 2011]. The *relevé* movement has been analyzed through the feature extracted to characterize the joint reaction forces and torque/moments at the ankle joint [Lin 2005] and from EMG data to assess motor control strategies [Masso 2004]. For the *pirouettes* movement, the quantitative technique is used to describe the whole-body angular momentum of the shoulder, arm, hip, and leg [Imura 2008] and the moment torque of the supporting leg [Imura 2010]. For the *élancer* movements, knee joint mechanics have been analyzed to determine the injury to the knee soft tissue [Kulig 2011], as has the estimation of muscle lengths for ballet dancers regarding injury in repetitive motion [Shan 2005]. For the *battement* movement, 2D video and digitization techniques were implemented to analyze pelvic movement during the *battement* movement [Deckert 2007].

From the previous discussion, the literature studies show that computerized instruments provide an effective means for quantitative analysis of human movement, in particular for the assessment of classical ballet dance. This is the motivation for the current work's development of a new dance training system that not only provides quantitative analysis of ballet dance but also automatically recognizes ballet gestures for assessment and virtual reality feedback, offering an effective method of enhancing dance learning.

2.2. Computational Techniques for Virtual Reality Dance Training

2.2.1. Instruments and Systems for Dance Training. Recently, there has been an increase in research interest in the development of a computer vision system for dance training and dance game entertainment. Various kinds of dance have been studied. These include Latin dance [Yang et al. 2012], hip hop dance [Yang et al. 2012], Agogo dance from Ghana [Deng et al. 2011], aerobics dance [Bobick et al. 2001], Shasa dance [Ho et al. 2013], ballroom dance [Uejou et al. 2011], and street dance [Naemura and Suzuki 2006]. As with the biomechanical research field, a majority of the research works in the analysis of human movement related to dance rely on 3D motion data, as with the works done in Raptis et al. [2011], Deng et al. [2011], Clay et al. [2009], Yang et al. [2012], Uejou et al. [2011], Bobick et al. [2001], Ho et al. [2013], and Alexiadis et al. [2011] (which employ a marker-based optical 3D motion capture system and Kinect system), whereas some works, as in Bobick et al. [2001] and Naemura and Suzuki [2006], employ a 2D image sequence. The general process for a dance training system starts with collecting data, followed by segmenting the dance sequence into meaningful sequences, recognizing the individual dance sequence, and giving feedback. However, to date, the research work that covers all aspects of this process at once is limited.

2.2.2. Recognition of Dance Gestures. Once the data for a full dance sequence has been collected, it can be analyzed into a series of movement phrases, which are in turn composed of linked dance gestures. To do this, we use a segmentation algorithm. Real-time segmentation is preferable to offline processing. The study in Barbic et al. [2004] shows that principal component analysis (PCA) and probabilistic PCA techniques are capable of offering online segmentation of motion capture data, according to high-level behavior (e.g., walking, running, punting). In comparison, the Gaussian mixture model (GMM) technique can be used as a batch process. For a dance sequence captured by video instruments, Bobick [2001] shows that the temporal template matching method is effective for temporal segmentation of dance. This method also addresses invariants to linear changes in speed and runs in real time.

Regarding the recognition task, the first main component is the feature set, which can be extracted from 2D and 3D data. The work in Schuldt et al. [2004] shows that local features extracted from the image sequence are robust to variations in scale, that is, the frequency and velocity of motion patterns for recognizing human motion, such as walking, running, jogging, and hand waving. These features can be transformed into a histogram of the class index of K-mean (i.e., bag-of-word technique) and passed to the SVM model for recognition. The work in Lv et al. [2005] constructs a motion template to represent a type of basic human action. Given the 3D joint position trajectories, a set of 2D data called motion channel is constructed to encode the evolution of a single joint coordinate for a specific action class. A weight parameter can be associated with this motion channel according to its discriminative power. Thus, this weight contributes to the effectiveness of the recognition stage, where a multiscale template matching is proposed to deal with possible temporal scale changes. The feature types represented in Schuldt et al. [2004] and Lv et al. [2005] are, however, only applied to basic human motion. The classifier in Lv et al. [2005] does not consider the joint distribution of motion features, which is important to identify complex dancing moves. In a dance sequence captured by the Kinect sensor, angular skeleton representation [Raptis et al. 2011] can be used to map the skeleton motion data to a smaller set of features, each of which is a scalar time sequence. The full torso is fit to a single frame of reference in order to compute the first- and second-degree limb joints. This process results in a feature set that is robust to noise, removing dependence on camera position and avoiding unstable parameter configurations such as near gimbal lock.

Measuring the similarity between two motion-captured data streams can be done by time-series techniques. A fixed similarity metric such as Euclidean distance, usually employed for this task, is inappropriate because of the inherent variability found in human motion. This problem can be overcome by a dynamic time warping (DTW) technique that aligns the time axis before calculating Euclidean distance [Keogh 2002]. However, it has been proven that DTW can only address the problem of local scaling and ignores global scaling. The latter scaling technique is very capable of solving the problem of variability in the speed of human motion [Keogh and Palpanas 2004]. Tang et al. [2008] proposed a similarity measure based on machine-learning techniques. The joint relative distance scheme is used as the basic feature. This is employed for training the system to compute the similarity of arbitrary motion pairs. When the skeleton data is characterized by a type of feature, a specifically designed classifier can be constructed. For example, based on the angular skeleton representation, a cascaded template matching is built for dance gesture classification [Raptis et al. 2001]. In training, the static model (prototype) is built for each gesture. In testing, the classifier correlates the input feature with the prototype gesture models and computes log-likelihood scores for each class. The winning match is then identified by ranking these scores, which is followed by performing rounds of logistic regression tests among the top classes.

In a VR system related to dance, recognition of gesture not only allows for retrieving the correct (teacher) gesture corresponding to an input student gesture but also allows the retrieved gesture to be used in the synthesis of the virtual dance partner. Deng et al. (2011) address the problem of real-time recognition of the user's live dance performance, to be used in determining the interactive motion to be executed by a virtual dance partner in an interactive dancing game. In that work, the partial encoding method is employed by first partitioning the human skeleton model into different body parts, each of which is then indexed by a separate SOM codebook and used for recognition. Such partial encoding has two advantages: (1) reducing the computational cost by partitioning whole-body motions with high dimension into a set of body-part motions of low dimension and (2) avoiding the disharmony that usually occurs between different body parts in dance.

2.2.3. Student Assessment and Feedback. In some dance training systems [Kavakli et al. 2004; Chua et al. 2003; Hachimura et al. 2004; Yang et al 2002], students can learn dance by watching videos or animations. Demonstrating and imitating the full dance at once is, however, not a practical way for people to learn because the whole dance may contain a lot of information for students to remember and learn [Yang et al. 2012]. Moreover, dance knowledge is acquired primarily through muscle learning and muscle memory, while watching a dance being demonstrated provides only faint meta-kinesthetic awareness, not the actual muscular involvement that performance does. The absence of the experience of muscular learning is addressed in the VR motion training system in Yang et al. [2002], which uses the idea of the "Ghost Metaphor," where the motion of the trainer in real time is superimposed on the trainee. The trainee observes the motion and follows the ghostly master to learn the motion. This system, however, can only give demonstration and cannot give any quantitative feedback to help students improve. Besides the superimposed method, Naemura and Suzuki [2006] also studied other basic visualization methods, which are face to face, face to face with mirror effects, and face to back. The results show that the superimposed method is the most effective for the repetition of partial movements, while the others are effective for whole movements. Some of these visualization methods are incorporated into our research. The present work offers visual feedback, providing the student assessments that can take the form of either a summative score or visual display that highlights the differences between the teacher's and the student's performance. In other words, the proposed system has the ability to sense the leaning task, thereby ensuring that the leaner's motions are captured and analyzed, and that the system provides trainees with feedback, notifying them how nearly their performance mirrored the teacher's.

To overcome the lack of feedback, some dance training systems work on quantitative measurement of a dancer's performance level [Chan 2011; Alexiadis et al. 2011; Naemura and Suzuki 2006; Raptis et al. 2011; Becker and Pentland 1996]. The work in Naemura and Suzuki [2006] associates motion features extracted from video with rhythm elements of dance action, which in turn shows a strong correlation with the subjective evaluation of performance levels. In Raptis et al. [2011], once the system identifies the best-matched class of human gesture, it examines how "well" the student performs this gesture compared with the teacher. DTW with exponential scaling of time-space is implemented to achieve the comparisons and obtains scores as an output. In the T'ai chi teaching system [Becker and Pentland 1996], the learner can play back and see the segment during which motion is most different from the expected motion for the gesture. The system then acts out the idealized motion of the gesture.

It is noted that the tasks for obtaining the score and visual feedback discussed here face difficulties stemming from the noise present in the skeleton data and the fact that humans exhibit a wide spectrum of ability to replicate a specific motion. This also makes it difficult to synchronize the student's character with the teacher's during the visual feedback in a manner that allows the student to differentiate the two motions. The visual feedback can be viewed as a motion synthesis problem, a data-driven animation where the motion-capture data can be used to control and direct a virtual character. This allows the student to immediately see his or her movement compared with the teacher's, using the simulated dancer. For instance, the motion synthesis system in Arikan and Forsyth [2002] uses a graph structure to effectively search for human motions that satisfy low-level user constraints, for example, a particular key pose in a particular time instant.

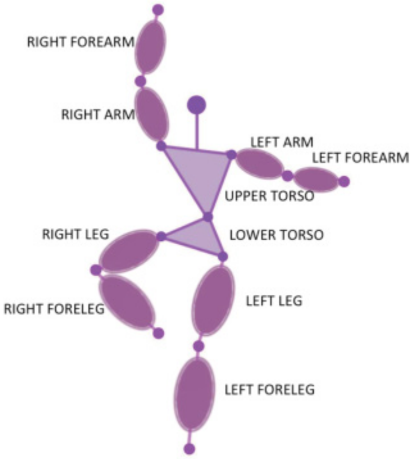


Fig. 2. Illustration of 10 segments of dancer's body.

Apart from the visual feedback method discussed earlier, a different strategy for teaching dance with feedback is to generate a dance lesson automatically according to learning objectives [Yang et al. 2012; Yang et al. 2013]. Here, instead of comparing student motion with the teacher models, the system uses the input dance sequence to automatically generate the lesson plan for students. The learning objectives are formed from the extracted dance patterns in order to further develop the knowledge structure. The system delivers the final output as a learning path to students who immediately see and need to mimic the teacher's moments.

3. FEATURE EXTRACTION

Elementary ballet training consists of the repetitive performance of a number of transitions from one basic posture (the body cuts a form in space, and the feet, limbs, hands, wrists, torso, shoulder, and head have harmonious relations with one another, which in elementary ballet are in large measure pre-established, being set by convention) to another and another (the sequence of transitions from posture to posture can be extended arbitrarily). The transitions from posture to posture must conform to a strict meter (the time interval between each posture/form in space and the next has to conform exactly to the requirements of the meter), and this rhythmic quality provides the basis for the evaluation of the sequence of transitions; further, the form that the body cuts in space (the disposition of the various parts of the body in relation to one another) must also closely approximate the ideal for that "posture" (in more advanced training methods, these forms in space become less conventional and instead reflect the imagination of the choreographer, who strives to create unique forms in space and imaginative forms of transition). We restrict ourselves to the more conventionally defined postures and transitions of elementary ballet and try to develop a ballet training system based on the recognition or movement patterns for teaching these elementary transitions—it is our intention that the system reflect necessities for assessing rhythmic precision of the performance and how closely the form the student's body cuts in space matches the idea.

The Microsoft Kinect system provides 20 three-dimensional skeleton points to represent each player (student) in the camera's field of view. To eliminate the invariance of the dancer's size and camera orientation, we obtain a set of 19 angle features from the 20×3 skeleton matrix for each person in a frame [Raptis 2011]. Figure 2 shows the

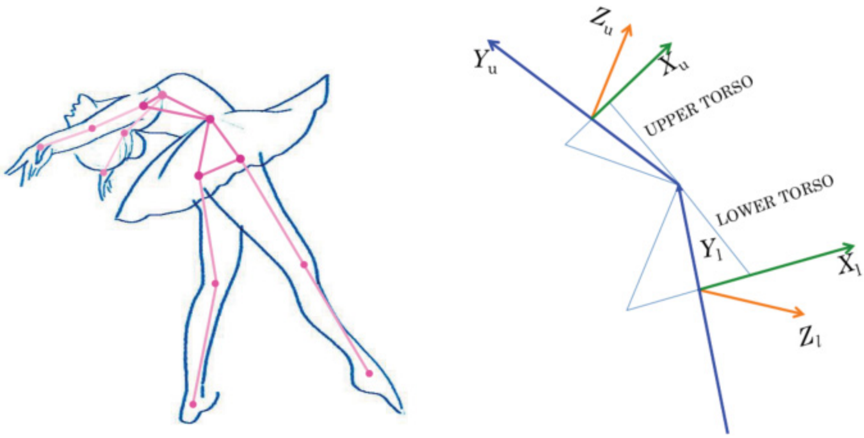


Fig. 3. The upper and lower coordinate systems extracted from dancer's torso. X_u, Y_u, and Z_u are the reference axes of the upper torso plane; X_l, Y_l, and Z_l are the reference axes of the lower torso plane.

10 parts of the human body used for feature extraction. We treat the torso as two parts: the upper torso and the lower torso. These will be used as the reference for other parts including the left/right arm, the left/right forearm, the left/right leg, and the left/right foreleg.

In performance, ballet dancers often bend, tip, and twist their torsos in various ways, many of which do not occur in daily activities. Identifying these variations is important for the accurate recognition of ballet gestures. The upper torso, which includes the spine joint and the left/right shoulder joints, can be treated as an upper foundation for other upper body segments. In the same way, the lower torso, which is made of the spine joint and the left/right hip joint, can also provide other lower body segments a reference coordinate system.

As illustrated in Figure 3, we build two 3D Cartesian coordinate systems based on the two sets of three joint points. In the upper torso system, the x-axis, X_u, is aligned with the line that connects the shoulders, oriented from left to right. We set the y-axis, Y_u, to be aligned with the line that is perpendicular to X_u. The z-axis, Z_u, of the orthonormal basis is also perpendicular to other two axes, and its direction is given by the right-hand rule, which is often used in vector cross-product. Accordingly, in the lower torso part, the x-axis, X_l, is aligned with the line that connects the left and right hip joint, and the axis orientation is also from left to right. The y-axis, Y_l, is aligned with the line that is perpendicular to X_l and must pass through the spine joint, which is shared between the upper and lower torsos. In both systems, the orientations of two y-axes pointing upward are canonically given. Finally, the z-axis, Z_l, in the lower system is also obtained from the right-hand rule.

As shown in Figure 4, we separately project the x-axis, y-axis, and z-axis of the upper torso coordinate system onto the $X_l - Z_l$, $X_l - Y_l$, and $Y_l - Z_l$ planes to obtain a provisional projected coordinate system $\{X'_u, Y'_u, Z'_u\}$. The orientation variance of the corresponding axis between $\{X_l, Y_l, Z_l\}$ and $\{X'_u, Y'_u, Z'_u\}$ coordinate system can represent the different status of the dancer's torso. Consequently, there will be three angles, α, β, and γ, which are angles between the x-axis, y-axis, and z-axis in these two coordinate systems, respectively. These angle features indicate the degree of twisting, tipping, and bending movement.

In this system, we use the upper and lower torso as references and measure other joint angles relative to these references. The first set of joints adjacent to the torso

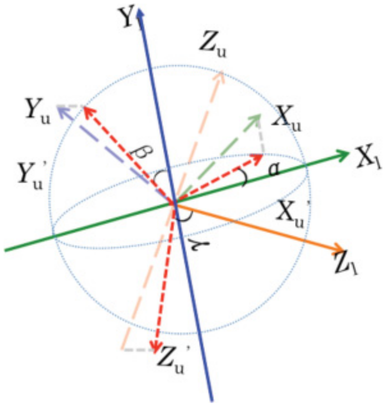


Fig. 4. Three angles features of torso: α, β, and γ.

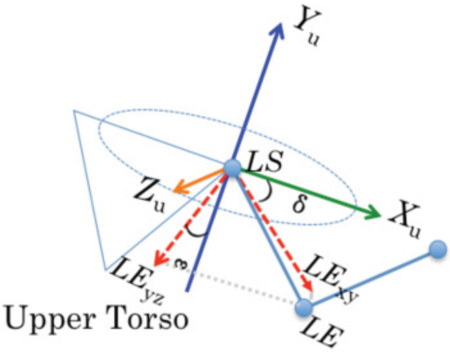


Fig. 5. Two angle features of left elbow (LE), δ and ε, in relation with the upper torso from the left shoulder (LS).

includes elbows and knees. We represent these points relative to the torso. The second set of joints includes the hands and the feet. Thus, there are in total eight joints that need to be translated into their corresponding upper and lower torso coordinate system.

To be precise, Figure 5 shows the feature extraction at the left elbow (LE). The two angles are calculated when the LE is projected onto the $X_u - Y_u$ and $Y_u - Z_u$ planes:

- Left–right swing δ—the angle between X_u and$\overrightarrow{(LS, LE_{xy})}$, where LE_{xy} is the projection of LE onto the $X_u - Y_u$ plane
- Forward–backward swing ε —the angle between Y_u and $\overrightarrow{(LS, LE_{yz})}$, where LE_{yz} is the projection of LE onto the $Y_u - Z_u$ plane

As a result, with this representation model, each joint is represented with two angles, $\{\delta, \varepsilon\}$. We denote the set of features obtained from skeletal frame as

$$\mathbf{f} = \{\alpha, \beta, \gamma, \delta_{LE}, \varepsilon_{LE}, \delta_{RE}, \varepsilon_{RE}, \delta_{LH}, \varepsilon_{LH}, \delta_{RH}, \varepsilon_{RH}, \delta_{LK}, \varepsilon_{LK}, \delta_{RK}, \varepsilon_{RK}, \delta_{LF}, \varepsilon_{LF}, \delta_{RF}, \varepsilon_{RF}\}, \tag{1}$$

where LE = left elbow, RE = right elbow, LH = left hand, RH = right hand, LK = left knee, RK = right knee, LF = left foot, and RF = right foot. We can also denote this set of features in time series as $\mathbf{f} = \{f_i(t), i = 1, \ldots, 19\}$ and emphasize the fact that we reduced the complexity of our input from a collection of 19 three-dimensional curves to a set of 19 one-dimensional vectors.

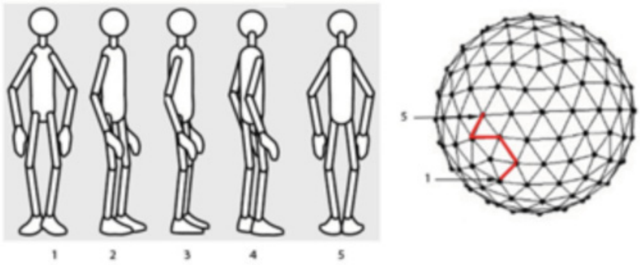


Fig. 6. Temporal sequence of postures representing an arbitrary movement. A simple "gesture" of five postures is displayed. The mapping of the gesture forms a trajectory on the SSOM in red.

In the experimental results reported in Section 6, we used the angle features described by Equation (1) for dance gesture recognition. As for comparison of recognition performance, we also used the normalized locations of all 20 joint positions, which are described by

$$\mathbf{l} = \{\hat{l}_i(t), i = 1, \ldots, 60\}, \tag{2}$$

where $\hat{l}_i$ is the location of the joint in one of the x/y/z planes. By considering all 20 joints in the three dimensions, the dimension of $\mathbf{l}$ will be 60. Here, the location $\hat{l}_i$ is normalized by considering the hip as the original location.

4. GESTURE RECOGNITION

The general approach taken in performing gesture recognition is to first automatically parse samples from across the spectrum of expected dance movements into a discrete set of postures (Section 4.1). This is achieved using an SSOM structure (Figure 6). In general, an SOM enables learned postures to be allocated to, and distributed across, nodes on a predefined lattice (map). The SSOM [Sangole et al. 2006], in particular, is well suited to distributing such postures so that they are separated in a maximal sense [Gonsales and Kyan 2012]. This property results from the wrap-around neighborhood learning that occurs when the lattice forms a closed-loop sphere. The utility of an SOM-based approach to parsing is that the discrete space is constructed in such a way as to retain associations that exist in the original input space; that is, postures (learned) are positioned in the map nearby to other postures that are very similar in nature. As a consequence of this topology-preserved mapping, a sequence of continuous postures (due to a movement or gesture) is expected to trace a rather smooth trajectory on the map. It is from this trajectory (sequence of key postures) that we formulate descriptors representing each gesture.

Due to variance in input posture sequences (sensor noise, inexact repetitions, etc.), multiple trials of the same gesture are projected onto the map, from which a *gesture template* may be constructed (Section 4.2). The goal of this template is to model the relative importance of certain postures within the gesture and also to promote generalization when detecting similar (but not exact) movements.

In the recognition stage, we consider an appropriate *matching* process, in which an unknown dance movement is associated with each gesture template, and the gesture class inferred. This process can be achieved in both an offline and online (real-time) context and will be discussed in Section 4.3. The purpose of recognition is to isolate or segment a continuous dance performance into the core set of linked gestural movements, which may then be compared quantitatively against known teacher movements (discussed in Section 5) in order to construct meaningful instruction and feedback on performance.

4.1. Building Posture Space

The construction of posture space amounts to the training of the SSOM using a random set of sample postures from the supervising (teacher) set of gesture movements. In practice, these samples are captured by Kinect (discussed in Section 6.1). It is important to note that the temporal order of posture samples is not important for building posture space; rather, the training set should provide a reasonable distribution over the range of postures that might be performed.

We define the training set $\boldsymbol{X}_t$ as the set of (teacher's) gestures $\boldsymbol{g}_{c,n}$:

$$\boldsymbol{X}_t = \{\boldsymbol{g}_{1,1}, \boldsymbol{g}_{1,2}, \boldsymbol{g}_{1,3}, \ldots, \boldsymbol{g}_{c,n}\}, \tag{3}$$

where $\boldsymbol{g}_{c,n}$ is the nth instance (recording) of gesture class c, and $\boldsymbol{x}^{\boldsymbol{t}}_{c,n} \in \Re^D$ is the posture feature vector of $\boldsymbol{g}_{c,n}$ at time t:

$$\boldsymbol{g}_{c,n} = [\boldsymbol{x}^1_{c,n}, \boldsymbol{x}^2_{c,n}, \ldots, \boldsymbol{x}^{\boldsymbol{t}}_{c,n}]. \tag{4}$$

The dimension of each posture feature vector D is dependant on the particular feature representation used. For joint positional data, where there are 20 individual 3D joints (x, y, z), $D = 60$, while for the proposed dance feature (discussed in Equation (1)), $D = 19$. Ultimately, nodes in the map (arranged into a spherical lattice) will compete via a learning mechanism (Algorithm 1) to represent input posture vectors from $\boldsymbol{X}_t$.

The map's spherical lattice is constructed by progressively subdividing a regular icosahedron down to a desired level (l). This results in a series of nodes uniformly arranged on a tessellated unit sphere (with uniform triangular elements). A sphere tessellated one level ($l = 1$) would result in 12 nodes, while ($l = 2$) and ($l = 3$) would each result in lattices of 42 and 162 nodes, respectively. Each node on the sphere is then represented by a weight vector $\boldsymbol{w}_{i,j,k} \in \Re^D$, which models a key posture from the input space $\boldsymbol{X}_t$. The total number of nodes represents the number of postures that can

ALGORITHM 1: Spherical Self-Organizing Map (SSOM)

```
input: map configuration (see Table I.)
output: weights for all nodes in the map $W$ ;

Initialize weights $\boldsymbol{w}_{i,j,k}$ (small random values)

repeat
    Get next input:           $\boldsymbol{x}^i$ = randomly select from training set $\boldsymbol{X}_t$
    Calculate node error:     $E^i_{i,j,k} = \varphi\left(u_{i,j,k}\right) \sum_{n=1}^{D} x^i_n - w_{n,i,j,k}$
    Select BMU:               $(i, j, k)^* = min\left\{E^i_{i,j,k}\right\}$
    Update BMU & neighbors:
    $\boldsymbol{w}_{(i,j,k)^*}(new) = \boldsymbol{w}_{(i,j,k)^*}(old) + \alpha\left[\boldsymbol{x}^i - \boldsymbol{w}_{(i,j,k)^*}(old)\right]$
                              where:
                              $\alpha = \mu\left(\frac{NE_{(i,j,k)^*}}{NE_{initial}}\right)$ = predefined learning rate
                              $NE_{(i,j,k)^*} = f\left(N_{cycle}\right)$ = neighborhood of BMU
                                        (decreases with $N_{cycle}$)
                              $NE_{initial}$ = initial neighborhood size (radius)
                              $\varphi\left(u_{i,j,k}\right)$ = count dependent, nondecreasing
                              function used to prevent cluster underutilization
    Increment $N_{cycle}$
until $N_{cycle} > Max\ Epochs$;
```

be learned by the map. In this representation, nodes are each equidistant from their immediate neighbors, with which they form a hexagonal neighbourhood.

The training phase of the SSOM is identical to the conventional 2D SOM. Given the input space $\boldsymbol{X}_t$ and the weight vectors $\boldsymbol{w}_{i,j,k}$, the system conducts learning according to the following process [Brennan et al. 2007]. First, the input posture vectors $\boldsymbol{x}^i$ are randomly introduced to the SSOM. For each input $\boldsymbol{x}^i$, the best matching unit (BMU) is selected. The BMU is the node on the map that is closest to the input $\boldsymbol{x}^i$ according to some similarity measure (e.g., L1 or L2 norm). In Algorithm 1, this similarity measure is denoted by $E^i_{i,j,k}$. We denote the BMU as $(i, j, k)^*$. Second, information from $\boldsymbol{x}^i$ is imparted to both the BMU node's weight vector $\boldsymbol{w}_{(i,j,k)^*}$ and the weight vectors in this node's immediate neighborhood on the map. This process of *information sharing* allows the map nodes to *tune* themselves to characteristic *postures* in the input space while forcing nearby nodes to tune to related or *adjacent* postures. Third, the same learning steps are repeated for remaining input vectors from the training set. As new input postures are presented from the training set, alternative BMUs compete for their representation, resulting in a locally organized distribution of key postures over nodes on the map. Finally, learning may be terminated after a fixed number of iterations or changes in node weights become negligible. In our case, we cease learning after a maximum number of iterations (cycles) have been reached.

4.2. Building Gesture Templates

Training of individual gesture templates involves projecting a set of labeled gesture sequences onto the learned posture space. For each posture sample from an input gesture, the projection involves finding the BMU and using this node to index the input sample. After projecting a temporal sequence of postures onto the map, an output sequence of indices results. The projection can be described as a sequence $\boldsymbol{s}_{c,n}$ of node indices or a trajectory $\boldsymbol{t}_{c,n}$ of individual node positions on the spherical surface (defined in a 3D coordinate system). As the sphere is of the unit radius, the trajectory can also be thought of as a sequence of 2D spherical coordinates.

In this work, we consider a number of alternative descriptors for a gesture instance and class given the sequence or trajectory traced on the SSOM:

(1) Posture Occurrence (PO)
(2) Posture Sparse Codes (PSCs)
(3) Posture Transitions (PTs)
(4) Posture Transition Sparse Codes (PTSCs)

4.2.1. Posture Occurrence (PO). PO is analogous to the popular bag-of-words (BOW) approach adopted in information retrieval (document and content-based image/video retrieval). In essence, each posture on the SSOM can be considered a unique word, while each gesture is a collection of individual words—structured according to a particular grammar (e.g., set, sequence, etc.). By aggregating the occurrence of postures in a gesture against the indexed set of nodes on the map, a histogram may be formed (over a single gesture or set of similar gestures), thus forming a template that may be used in recognition. In this first descriptor, a histogram is formed for each instance n in the teacher's gesture sequence $H_{c,n} = hist(\boldsymbol{s}_{c,n})$. A template histogram for the gesture class may also be formed by summing over the set of $H_{c,n}$, where $c = 1, 2, \ldots, K$ represents the set of gesture classes:

$$PO_c = \frac{\sum_{n=1}^{N} H_{c,n}}{\left|\sum_{n=1}^{N} H_{c,n}\right|}. \tag{5}$$

4.2.2. Posture Sparse Codes (PSCs). PSCs are similar to posture occurrence histograms; however, the sparse code only represents the existence of a set of postures, and not their frequency of occurrence. For instance, if a particular gesture involves a set of five postures, some of which are held for a length of time, then the sparse code will only indicate that they occurred and won't consider the duration. This offers a time-invariant measure of posture existence and is useful when detecting gestures that may be performed at different speeds. The sparse code can be obtained from $H_{c,n}$:

$$SC_{c,n}(i) = \begin{cases} 1, & if\ H_{c,n}(i) > 0 \\ 0, & otherwise \end{cases}. \tag{6}$$

One might then consider a histogram of sparse codes PSC_C over the set of gesture instances as a representation for each gesture class template:

$$PSC_c = \frac{\sum_{n=1}^{N} SC_{c,n}}{\left|\sum_{n=1}^{N} SC_{c,n}\right|}. \tag{7}$$

4.2.3. Posture Transitions (PTs). The first two templates do not consider the temporal arrangement of postures in the map. In the same way that individual nodes in the map are indexed, so too are pairs of postures from the map. By forming histograms over the occurrence of transitions between any two postures in the map, a template is constructed that considers a *bag of segments*: collating partial sequences from within the gesture, thereby incorporating temporal aspects of the gesture trajectory through posture space. The descriptor allows for some generalization by not requiring strict adherence to an exact sequence over the whole gesture, but rather, it emphasizes *shared partial* sequences that co-occur across gesture samples in the training set. Posture transitions can be generated using an adjacency matrix $\mathbf{A}_{ij}$, where the i, jth entry represents the occurrence of the transition from postures $i \rightarrow j$ in the indexed set of map nodes. Occurrences are aggregated by passing a sliding window (size = 2) over the posture sequence $\boldsymbol{s}_{c,n}$.

4.2.4. Posture Transition Sparse Codes (PTSCs). PTSCs are analogous to sparse codes of postures, only they represent the existence of transitions rather than the frequency of transitions. In the definition of *PTSCs*, it is possible for there to be transitions from a node to itself (i.e., when a movement remains in a given posture for a brief period). By considering the sparse code of these transitions, extended periods within the same posture will not dominate the descriptor.

4.3. Recognition Framework

In order to perform matching between an incoming gesture $\boldsymbol{g}_u$ and known templates (discussed in Section 4.2), the incoming set of postures is projected onto the SSOM to extract the unknown posture sequence $\boldsymbol{s}_u$. This projection may be conducted offline (after the student has performed a set of moves) or online (as the student is performing a set of moves).

In either case, the task of recognition is nontrivial, due to the differing lengths of gestures (across classes) and the differing speeds with which they may be enacted (by the student/teacher). In order to address this, we propose an online probabilistic framework (inspired by the work of Kawashima et al. [2009]). Like Kawashima et al., we adopt a simple Bayesian framework for progressively estimating an updated posterior probability $P(c|\boldsymbol{s}_u)$ for each of the $c = 1 \dots K$ gesture classes. In the work of Kawashima et al., the likelihood is computed at each unit of time by considering the single posture triggered on the map and whether or not it occurred in each gesture template. The likelihood $P(\boldsymbol{s}_u|c)$ was computed as the ratio of the existence of the current posture in gesture c to the total number of different postures in c. In this work, we reframe the likelihood as a histogram intersection (Equation (11)) between a progressively growing sequence $\boldsymbol{s}_u$ (inclusive of postures from time t_0 to t), which may be described as a histogram of either *PO*, *PSC*, *PT,* or *PTSC* (defined in Section 4.2), versus the corresponding template histograms for each gesture class.

In this framework, we consider h_s to be the histogram feature for the current sample at time t, and h_c to be the histogram template for class c. We thus define (for time t) the posterior $P_t(c|h_s)$, likelihood $P_t(h_s|c)$, and prior $P_t(c)$ probabilities according to the following:

$$P_t(c|h_s) = \frac{P_t(h_s|c)P_t(c)}{P_t(h_s)} = \frac{P_t(h_s|c)P_t(c)}{\sum P_t(h_s|c)P_t(c)} \tag{8}$$

$$P_t(h_s|c) = HI(h_s, h_c) \tag{9}$$

$$P_t(c) = \begin{cases} \dfrac{1}{K}, & if\ t = t_0 \\ \dfrac{P_{t-1}(c|h_s) \cdot HI(h_s, h_c)}{\sum_K P_{t-1}(c|h_s) \cdot HI(h_s, h_c)}, & otherwise \end{cases} \tag{10}$$

$$HI(h_s, h_c) = 1 - \sum_i \min[h_{s,i}, h_{c,i}]. \tag{11}$$

The mechanism for inferring the appropriate gesture class is summarized in Algorithm 2. According to the previous equations, the input sequence is allowed to accumulate postures over time t, where for each instant, the accumulated gesture is projected onto the SSOM to generate a posture sequence, which can be converted into one of the four histogram representations from Section 4.2. Likelihoods are estimated as histogram intersections (Equation (11)) between each template histogram and that computed from the input posture sequence. A perfect intersection with a template will yield a likelihood of 1 for a given class. It is important to note that all histograms are normalized (even if calculated from a gesture sequence containing only a single posture).

As the sequence begins to resemble a gesture from the known set, its posterior will grow and eventually surpass a detection threshold T. Upon triggering this threshold, the class c with the maximum posterior is considered detected, and the system resets the priors for all classes and recalculates the posterior. At this point, in order to free up postures from the accumulated sequence, t_0 is set to the current time; thus, the newly considered sequence grows again from this instant (flushing all past postures). This process continues, triggering new instances of detected gestures, until the end of the input sequence is reached (or in the case of online detection, the system stops acquiring input posture data).

5. DANCE VISUALIZATION AND USER FEEDBACK

Typically, there will be three stages in the student's interaction with the system. First, the student will watch as a virtual teacher demonstrates a gesture (this step in the process will be driven by gesture data from the database). Second, the student will attempt to repeat the dynamic phrase the virtual teacher just performed. When the student has completed his or her performance, the system gesture recognition function will be activated, and the gesture data for the teacher's performance will be the corresponding gesture called up and sent to the system's feedback component. Finally, the

ALGORITHM 2: Gesture Recognition Using Histogram Intersection

```
input: gesture sequence $\boldsymbol{g}_u = \left[\boldsymbol{x}^{t0}, \ldots, \boldsymbol{x}^{t}, \ldots, \boldsymbol{x}^{T}\right]$
output: $P_t\,(c|h_s)$; $argmax_{c,t}\,\{P_t\,(c|h_s)\}$

Set $t = t_0 = 0$;
repeat
        Let input gesture $\boldsymbol{g}_u = \left[\boldsymbol{x}^{t0}, \ldots, \boldsymbol{x}^{t}\right]$
        Calculate the likelihood $P_t\,(h_s|c)$ using Equation (9)
        Calculate the prior $P_t\,(c)$ using Equation (10)
        Calculate the posterior probability $P_t\,(c|h_s)$ for all c using Equation (8).

        If $max\left[P_t\,(c|h_s)\right] > \mathrm{T}$ (let T = threshold)
            $t_0 = t$
            reset prior $P_t\,(c) = {}^{1}\!/_{K}$
            recalculate posterior $P_t\,(c|h_s)$

        $t$++
until $t > T$ (end of input sequence);
```

student will watch the feedback provided to him or her in an immersive 3D environment. The feedback will furnish the student with information about how closely his or her performance imitated the teacher's.

After the dance element as performed by the student is recognized, an immersive visual feedback based on a VR environment will be used to allow students to examine the differences between their performance and the teacher's and to discern which parts of their performance most need improvement. In the CAVE system, the teacher dancer and the student will do their performances in a full 3D environment as often as needed and, at the same time, can allow the student to view their performances from the audience's vantage point and present to the student a real-time analysis of the performances. The following three types of *feedback* and two types of *playback* are provided in our visual feedback subsystem.

5.1. Feedback Mode

A metronome is employed to synchronize the time-series motion data between student and teacher. The dancer usually performs dance according to his or her "internal clock" [McAuley et al. 2003]. An assumption can be made that the most likely perceived beat of the metronome for a rhythmic pattern is based on the beat match of an internal clock of the dancer. Ideally, the timings when the dancer stretches out or draws in his or her limbs must, to an extent, match the rhythm of the metronome. However, this does not take into account the variability in the speed of human motion; for example, a dancer may perform a fast or slow segment of his or her body part. The DTW can be adopted to handle timing variation, since it can offer the solution for the time-alignment problem of the time-series signal [Raptis et al. 2011; Keogh et al. 2004]. Thus, successful synchronization will allow better visual feedback and meaningful scores when comparing student performance with the teacher.

Once the system identifies the best-matched gesture class, the question remains as to how well the student is able to perform this gesture compared to the teacher. The feedback methods are described as follows:

5.1.1. Side by Side. Virtual models will play back the most recent performance of the student and the teacher side by side, each in its own half of the screen (see Figure 7(b)).

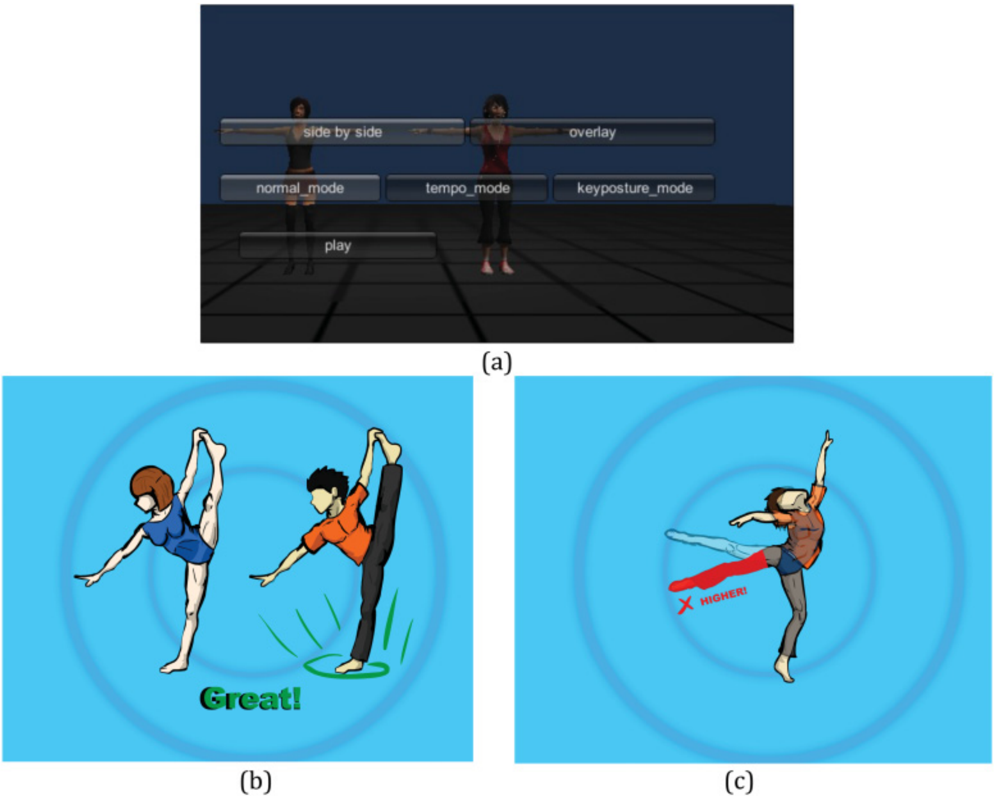


Fig. 7. (a) The control panel of visualization part. (b) Side-by-side feedback. (c) Overlay feedback.

The system also provides for the student and the teacher models to face each other while performing this dance element, to have their backs to the audience, or to stand facing the audience, though the last is generally the most useful. When the student is in the CAVE, with the "magic wand," he or she can adjust the angle of view on the teacher's performance and examine how it appears from two alternative vantage points.

5.1.2. Overlay. A second way of playing back the student's most recent performance of a dance element involves overlaying the student's performance on the teacher's. As shown in Figure 5(c), because the virtual student's appearance is somewhat different from the teacher's, after the alpha value for the virtual student is set to 0 to 0.5, it is easy for a viewer to distinguish the virtual student's performance from the teacher's and to see where the two diverge (indicating areas the student must work on).

5.1.3. Score Graph. The student's performance is scored and the score presented to student in the form of either a number or a curve (trace). After setting the time alignment of the time-series data, the value of the curve at point t is calculated as follows:

$$Score_t = 1 - d_t, \tag{12}$$

where d_{t} is the relative distance at time t between the student's features and teacher's:

$$d_t = \sum_{i=1}^{N} \left| \frac{\theta_{si} - \theta_i}{\max\limits_i \theta_i - \min\limits_i \theta_i} \right|, \tag{13}$$

where θ_i and θ_{si} are the ith feature angles of the teacher and the student, respectively, and N is the total number of features. Thus, the curve allows the student to see how closely his or her performance resembles the teacher's across the duration of the performance—to see, then, by the development of the curve, where the performances diverge and where they converge. When the similarity measure is less than a predefined threshold, the curve turns red.

5.2. Playback Mode

One key element in ballet training is the synchronization of movements to music. Kassing and Jay [1998] stress the importance of precision in ballet by saying that students must learn to be at a certain place on a certain beat. The beginning and endings of dance phrases have particular importance. We will call the relative arrangements of the body parts at these highly significant moments "key postures." To help reinforce the importance of these key moments, we have developed three playback modes, which are normal mode, key posture mode, and tempo mode.

Normal Mode: The feedback will be display normally at the rate of 30fps without pause. *Key-Posture Mode*: In this mode, the virtual student's and teacher's performance are all supposed to match the cadence points in the music. The performance can be halted for up to 2 seconds at every beat point to allow the divergences between two key postures to be examined. After 2 seconds have expired, both performances begin in sync. *Tempo Mode*: Among the key problems in teaching beginning students dance is to get them to perform "on the beat"—that is, to develop the facility to time the key postures so they occur at cadence points in the music and to perform the dance elements rhythmically, giving its various phases harmonic relations with one another. The performance of the virtual teacher and student can be paused separately at his or her corresponding key postures. Because the teacher's key postures are supposed exactly on beats, the teacher still will be paused at its beat. However, the student's key posture may be advanced or delayed relative to the teacher's performance, and the student would know how he or she could adjust his or her speed to correct timing errors. After they are all paused for 2 seconds, the two sequences will continue to play back together.

6. PERFORMANCE EVALUATION

The proposed system outlined in Figure 1 was implemented. The CAVE has four stereoscopic projectors and screens correspondingly. Driven by a graphics cluster of five nodes, one node serves as the cluster master while the other four drive the corresponding screens. The user wears active stereo glasses containing targets of several light refraction markers in a fixed geometry. The location and orientation of the user's eyes are traced by a 6-degree-of-freedom (6DOF) tracking system. A tracking server calculates each target's position and orientation based on images captured by tracking cameras distributed on top of the screens. The tracking data is used to determine the content to be displayed on the screens. We used the 3D Unity game engine and visual C# to implement the feedback engine and interface with the Kinect sensor. MiddleVR was used to control the graphics in the CAVE.

Table I. SSOM Training Configurations Considered

Parameters	C0	C1	C2a	C2b	C3a	C3b
Icosahedron level	0	1	2	2	3	3
Map nodes	12	42	162	162	642	642
Neighborhood(s)	2	3	4	6	4	6
Epochs	50	50	100	100	100	100

Table II. Isolated Gesture Database

Ballet Gesture		# Instances (# Frames)	
Label	Description	Teacher	Student
G1	1st Position → 2nd Position	8 (56–96)	5 (50–62)
G2	2nd Position → 3rd Position	10 (57–77)	5 (35–62)
G3	3rd Position → 4th Position	8 (59–75)	5 (51–58)
G4	4th Position → 5th Position	10 (48–81)	5 (49–74)
G5	5th Position → 6th Position	10 (43–80)	5 (58–74)
G6	6th Position → 1st Position	10 (41–88)	5 (46–65)

Table III. Continuous Gesture Database

Ballet Dance		# Instances (# Frames)	
Label	Postures (Gesture Sequence)	Teacher	Student
D1	Rest→1st→2nd→3rd→4th→5th→Rest (G6,G1,G2,G3,G4,G5)	1 (281)	1 (273)
D2	Rest→1st→5th→4th→3rd→1st→Rest (G6, XX*, ~G4, ~G2, XX, ~G6)	1 (270) 1 (270)	1 (277) 1 (277)

* XX = no representation.
~ Indicates the reversal of a gesture.

6.1. Gestural Subsystem Configuration and Test Data

A number of configurations were considered for the training of posture space (Table I). All maps were trained according to two input spaces (for the joint position described by Equation (2) and angle features described by Equation (1), respectively). For brevity, we restrict our discussion to the analysis of results pertaining to SSOM configuration C2a.

In training gestural trajectories, and as a proof of principle for the proposed framework, two datasets were constructed: one representing Teacher (used for both construction and testing of gesture recognition performance) and one representing Student (used for testing only). The databases include a set of six isolated gestures (i.e., each gesture G1 to G6 is recorded individually, independent of any sequence of other movements/gestures). The structure of this database is summarized in Table II. Figure 8 shows the six basic positions (i.e., six postures) of ballet dance. Gesture G1 is the dance gesture moving from the first position to the second position. All gestures, G1 to G6, are defined in Table II. In order to assess the online capability of the system to recognize and isolate gestures from a continuous dance sequence, a second database was constructed (Table III). In this, recordings were collected for two different sequences of dance movements (D1 and D2). Again, for brevity, we restrict our attention to D1 (for which all component gestures have a representation in our trained posture space).

6.2. Stability of Posture Space Projections

In the first set of experiments, observations are made as to how the variability in repeated gestures maps into posture space. Figure 9 shows a series of mappings of

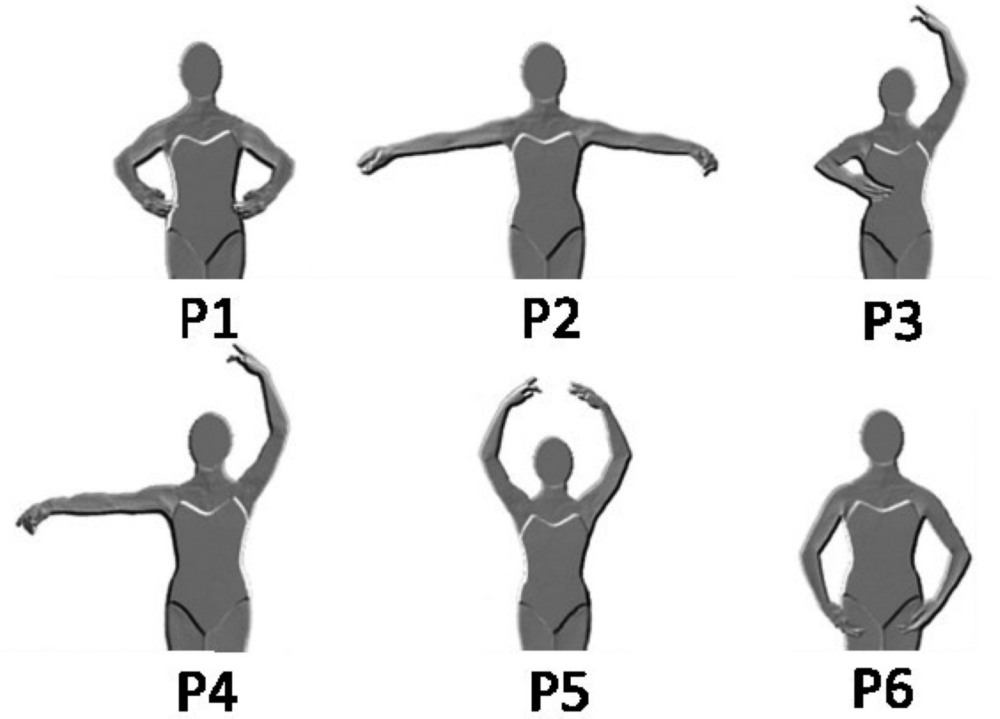


Fig. 8. Six basic positions in ballet dance. First position (posture 1): both arms lifted in front. Second position (posture 2): both arms opened. Third position (posture 3): left arm in front and right arm lifted up. Fourth position (posture 4): left arm open to the side and right arm lifted up. Fifth position (posture 5): both arms lifted up. Sixth position (posture 6): both arms put down.

gesture instances (columns) per gesture type (rows). A visualization of the SSOM and associated gesture trajectories shows that even differences in frame length and duration of the gesture (variations of up to 40% difference in frame length) do not appear to impact the consistency with which the gesture maps onto posture space. All gestures appear to trace quite characteristic and repeatable paths on the unit sphere. The start (solid blue marker) and end points (solid red marker) of the trajectories are also shown. Although gestures G5 and G6 are quite similar in terms of the postures traced, there is quite a clear difference in the direction of the trajectory.

In Figure 10, trajectories are mapped using the dance feature. Again, the repeated patterns on the SSOM indicate consistent mapping into posture space. In this case, it would appear that the dance feature traces quite a wide trajectory (relative to the joint position feature), using up much more of the sphere. It is clear from these mappings that the paths traced for different gestures are quite unique from one another, which is expected to translate into better discrimination between trajectories (and therefore gestures).

In both scenarios, the consistency of the mapping indicates some stability in the representation of gestures and suggests that sufficient overlap should exist when generating histogram templates.

6.3. Static Gesture Evaluation

In order to assess the performance of the SSOM posture space representations, gesture template definitions, and matching criteria, a number of trials were conducted on the

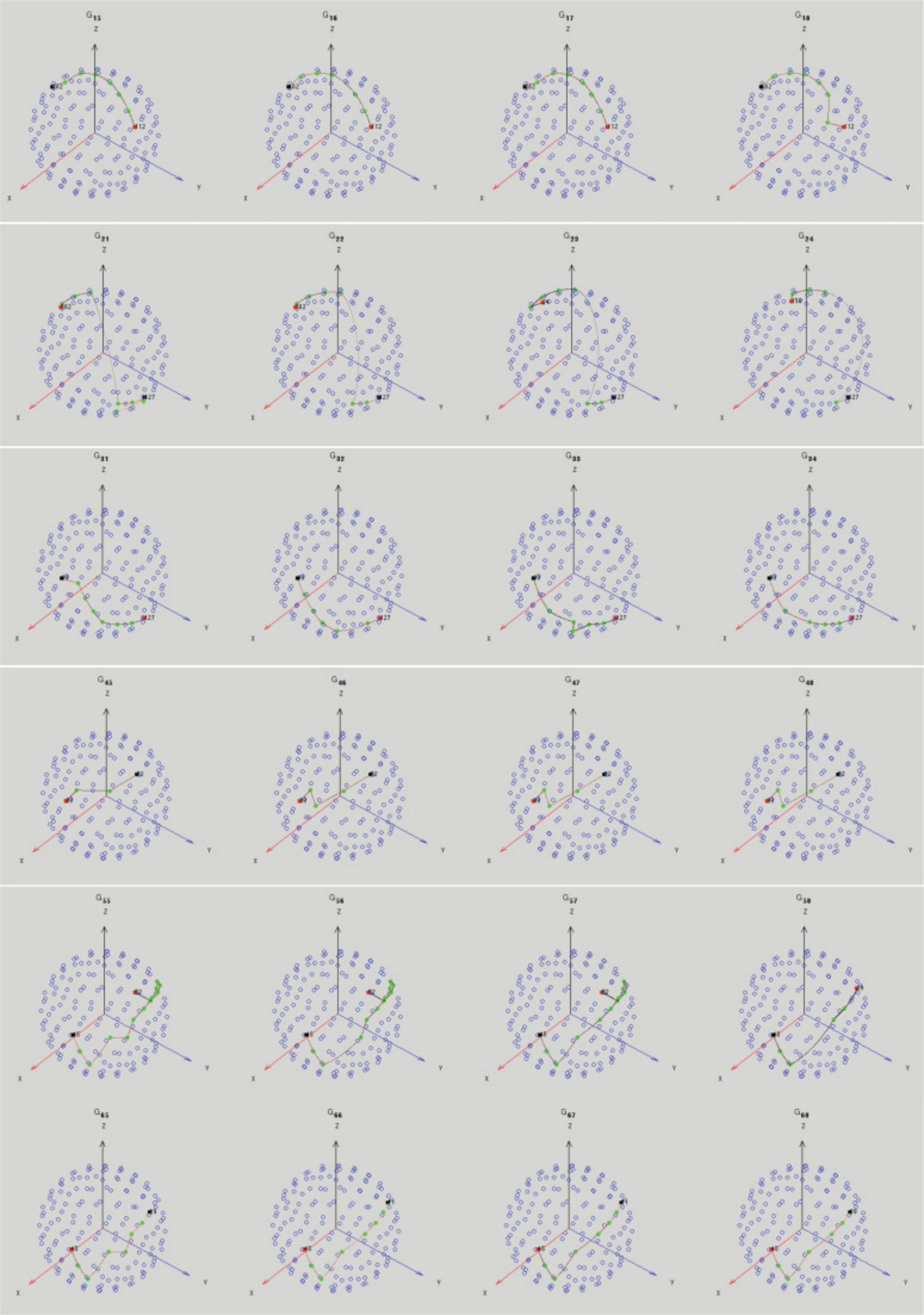


Fig. 9. Gesture projections (joint position): instances of gestures G1 to G6 (rows top to bottom), respectively. Smooth, local sets of postures show stable, highly repeatable trajectories. Note: G5 and G6 include similar postures with opposing trajectory paths.

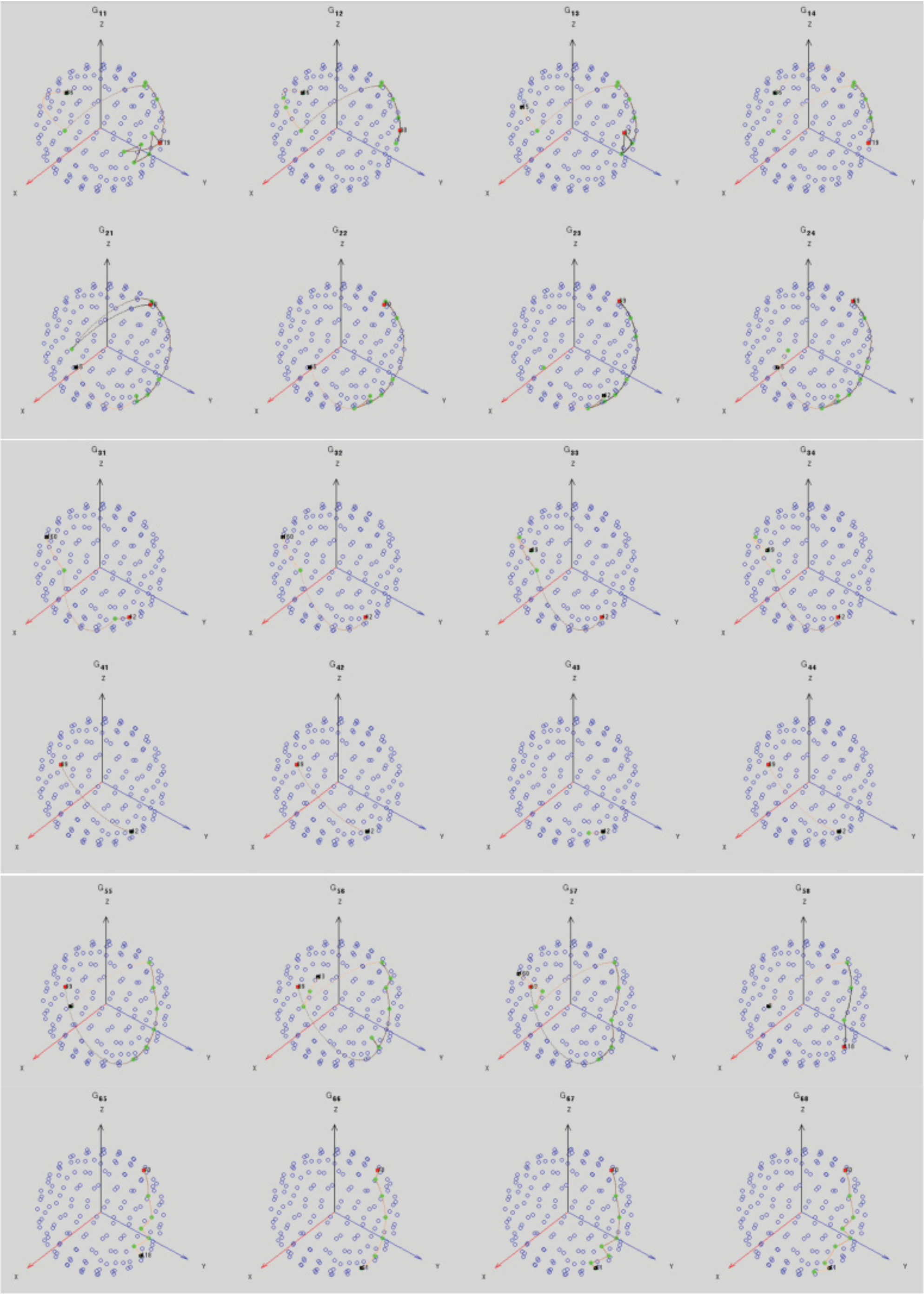


Fig. 10. Gesture projections (dance feature): instances of gestures G1 to G6 (top to bottom, respectively). Trajectories share many postures and tend to trace over a broader region on the map.

Table IV. Isolated Gesture Recognition – Posture Occurrence (SSOM: C2a; Teacher)

	Average Recognition Accuracy% (10 Trials) (20% Train, 80% Test)						Average Recognition Accuracy% (10 Trials) (40% Train, 60% Test)					
Gesture	Joint Positions			Dance Angles			Joint Positions			Dance Angles		
	L1	L2	HI	L1	L2	HI	L1	L2	HI	L1	L2	HI
G1	100	100	100	100	100	100	100	100	100	100	100	100
G2	97	93	97	100	100	100	98	95	98	100	100	100
G3	100	100	100	100	100	100	100	100	100	100	100	100
G4	100	100	100	79	64	79	100	100	100	91	90	91
G5	98	95	98	90	80	90	100	99	100	91	91	91
G6	98	92	98	100	99	100	99	96	99	100	100	100
Overall	98.8	96.7	98.8	94.8	90.5	94.8	99.5	98.3	99.5	97.0	96.8	97.0

Table V. Isolated Gesture Recognition – Posture Sparse Codes (SSOM: C2a; Teacher)

	Average Recognition Accuracy% (10 Trials) (20% Train, 80% Test)						Average Recognition Accuracy% (10 Trials) (40% Train, 60% Test)					
Gesture	Joint Positions			Dance Angles			Joint Positions			Dance Angles		
	L1	L2	HI	L1	L2	HI	L1	L2	HI	L1	L2	HI
G1	100	100	100	100	100	100	100	100	100	100	100	100
G2	100	100	100	100	100	100	100	100	100	99	99	99
G3	92.5	92.5	92.5	100	100	100	100	100	100	100	100	100
G4	92	92	92.5	96	90	96	100	100	100	100	98	100
G5	100	100	100	80	90	81	100	100	100	81	92	82
G6	100	100	100	100	100	100	100	100	100	100	100	100
Overall	97.4	97.4	97.4	96.0	96.7	96.2	100	100	100	96.7	98.2	96.8

isolated gesture database (Teacher). In this section, two test cases were considered: the first using a (20%:80%) ratio of training samples to test samples and the second using a (40%:60%) ratio. In other words, from the full set of Teacher gestures, 20% (e.g., two of 10 instances from each gesture) were randomly selected and used to form gesture templates, while the remaining 80% were classified against those templates. This process was repeated for 10 trials, and the accuracy of classification was recorded per class (for each set of input features and similarity metrics). Here, the template matching was performed by three similarity metrics: L1 norm, L2 norm, and histogram intersection. The L1 norm is defined as

$$|\boldsymbol{h}_s, \boldsymbol{h}_c|_1 = \sum_{i=1}^{D} |h_{s,i} - h_{c,i}|. \tag{14}$$

The L2 norm is defined as

$$|\boldsymbol{h}_s, \boldsymbol{h}_c|_2 = \sqrt{\left(\sum_{i=1}^{D} \left|h_{s,i} - h_{c,i}\right|^2\right)}. \tag{15}$$

Histogram intersection is defined by Equation (11). The second experiment followed the same process but instead using 40% of samples to generate gesture templates, with 60% used for classification. The results are displayed in Tables IV to VII.

From this data, for G1, G2, G3, and G6 (and with the exception of highlighted angles), the dance feature angle appears to be more robust over several trials (20% and 40% training set, respectively). It would appear that there are some discrepancies in the angle calculations that are causing noise for gestures G4 and G5. For gestures G1 to G3 and G6, the dance angles perform better overall than the straight joint position

Table VI. Isolated Gesture Recognition – Posture Transitions (SSOM: C2a; Teacher)

	Average Recognition Accuracy% (10 Trials) (20% Train, 80% Test)						Average Recognition Accuracy% (10 Trials) (40% Train, 60% Test)					
Gesture	Joint Positions			Dance Angles			Joint Positions			Dance Angles		
	L1	L2	HI	L1	L2	HI	L1	L2	HI	L1	L2	HI
G1	100	100	100	100	100	100	100	100	100	100	100	100
G2	100	100	100	100	100	100	97	94	97	100	99	100
G3	92.5	92.5	92.5	100	100	100	100	100	100	100	100	100
G4	92	92	92.5	79	64	79	100	100	100	91	90	91
G5	100	100	100	88	80	88	100	97	100	91	89	91
G6	100	100	100	100	99	100	100	97	100	100	100	100
Overall	97.4	97.4	97.4	94.5	90.5	94.5	99.5	98	99.5	97.0	96.3	97.0

Table VII. Isolated Gesture Recognition – Posture Transition Sparse Codes (SSOM: C2a; Teacher)

	Average Recognition Accuracy% (10 Trials) (20% Train, 80% Test)						Average Recognition Accuracy% (10 Trials) (40% Train, 60% Test)					
Gesture	Joint Positions			Dance Angles			Joint Positions			Dance Angles		
	L1	L2	HI	L1	L2	HI	L1	L2	HI	L1	L2	HI
G1	100	100	100	100	100	100	100	100	100	100	100	100
G2	100	100	100	100	100	100	100	100	100	100	100	100
G3	92.5	92.5	92.5	100	100	100	100	100	100	100	100	100
G4	92	92	92.5	96	92	96	100	100	100	100	100	100
G5	100	100	100	87	89	87	100	100	100	89	90	89
G6	100	100	100	100	100	100	100	100	100	100	100	100
Overall	97.4	97.4	97.4	97.2	96.8	97.2	100	100	100	98.2	98.3	98.2

feature. Having said this, recognition performance is quite high for all scenarios, in part due to the simplicity of the gesture movements recorded.

If considering the noisy data in these “problematic” gestures, it seems evident that the sparse code histograms (of either posture occurrence or posture transition) appear to improve recognition performance. In addition, sparse codes of posture transitions appear to give the best performance overall. It should be noted that the approaches based on posture transitions (or their sparse codes) consider temporal information from the gesture, so it would seem justified that performance is improved.

The performance of joint positions alone cannot be discounted, although this may be due to the simple motions conveyed by the gesture set. It would seem plausible that a positional feature might be suitable in a multiresolution framework, for either filtering out coarse-grained body postures or filtering/constraining recognition to an appropriate subset of gestures, when a large-scale set of complex gestures is to be recognized. For fine-scale recognition or recognition of more complex movements, the dance angle feature proposed seems appropriate.

6.4. Generalization Performance in Gesture Recognition

In this experiment, we attempt to assess whether the methods we have developed for the recognition of our basic set of gestures in performance can be generalized to an expanded set of gestures. Based on the six postures discussed previously (see Figure 8), we define a new set of gestures, Set I, which contains a total of N gestures. Here, $N = \sum_{p=1}^{P} (p - 1)$, where $P = 6$ is the total number of postures. Table VIII shows a matrix describing the definition of all gestures. In the table, giving the six postures P_1 to P_6, the gesture G_{ij} is formed as an isolated gesture moving from the ith position

Table VIII. Definition of the 30 Gestures

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	-	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}
P_2	G_{21}	-	G_{23}	G_{24}	G_{25}	G_{26}
P_3	G_{31}	G_{32}	-	G_{34}	G_{35}	G_{36}
P_4	G_{41}	G_{42}	G_{43}	-	G_{45}	G_{46}
P_5	G_{51}	G_{52}	G_{53}	G_{54}	-	G_{56}
P_6	G_{61}	G_{62}	G_{63}	G_{64}	G_{65}	

Note: P_i is the *i*th posture. G_{ij} is the gesture performed from the *i*th posture to the *j*th posture. G_{ij} is the reversal of the gesture G_{ji}.

Table IX. Isolated Gesture Database

	# Instance for Each Gesture			
Gesture	Teacher	Student1	Student2	Total Instances
Gesture Set I: G_{12}, G_{13}, G_{14}, G_{15}, G_{16}, G_{23}, G_{24}, G_{25}, G_{26}, G_{34}, G_{35}, G_{36}, G_{45}, G_{46}, G_{56}	10	10	10	450
Gesture Set II: G_{21}, G_{31}, G_{41}, G_{51}, G_{61}, G_{32}, G_{42}, G_{52}, G_{62}, G_{43}, G_{53}, G_{63}, G_{54}, G_{64}, G_{65}	10	10	10	450

to the *j*th position (i.e., moving from posture P_i to posture P_j). This definition forms the gesture set, Set I, in the upper triangle of the matrix, containing $G_{12}, \ldots, G_{16}$; $G_{23}, \ldots, G_{26}$; $\ldots, G_{56}$, which has a total of $N = 15$ gestures. By contrast, the gesture G_{ji} is the reversal of the gesture G_{ij}. The reversal gestures form the gesture Set II, which contains gestures in the lower triangle of the matrix. The total number of gestures is obtained from the union of Set I and Set II, which contains $2 \times N = 30$ gestures.

We first used the nonreversal gestures in Set I. Three datasets were constructed: Teacher dataset, Student1 dataset, and Student2 dataset. The Teacher dataset and Student1 dataset were used for both construction and testing of gesture recognition performance, whereas the Student2 dataset was used for testing only. Thus, the Student2 dataset is considered as unseen data to the trained system. The database includes 15 isolated gestures (i.e., each gesture is recorded independently of any sequence of other movement/gesture). The structure of this dataset is summarized in Table IX. In order to assess the performance of the SSOM posture space representation, gesture template definitions, and matching criteria, the system was trained by a (50%:100%) ratio of training samples. From the full set of Teacher gestures and Student1 gestures, 50% (e.g., 10 of 20 instances from each gesture) were randomly selected and used to form gesture templates, while all 100% were classified against these templates. This system employed the SSOM configuration C2a and trained according to the joint position feature.

Table X shows the performance of the proposed system for recognition of ballet dance performed by three people, Teacher, Student1, and Student2. The system can attain more than a 98% recognition rate averaged over 15 classes for recognition of the Teacher dataset by using the PT template and histogram intersection (HI) for similarity matching. The PO template also gave similar recognition performance to the PT method. Moreover, the system can recognize dance from the Student1 dataset at 100% accuracy by using the PO template and L2 norm for similarity matching. It can also be observed that by using other students' data for testing (e.g., the Student2 dataset), the proposed system can still achieve the average recognition accuracy as high

Table X. Gesture Recognition Results Averaged over 15 Gestures Defined in the Upper Triangle in Table VIII

Testing Data	Descriptor	Average Recognition Accuracy (%)			Testing Data	Descriptor	Average Recognition Accuracy (%)		
		L1	L2	HI			L1	L2	HI
Teacher	PO	96.7	98.0	96.7	Student2 *(unseen data)*	PO	90.7	**92.0**	90.7
	PSC	79.3	84.0	79.3		PSC	69.3	72.0	69.3
	PT	98.7	97.3	**98.7**		PT	91.3	88.7	91.3
	PTSC	87.3	92.7	87.3		PTSC	67.3	73.3	67.3
Student1	PO	94.0	**100**	94.0					
	PSC	77.3	85.3	77.3					
	PT	94.7	99.3	94.7					
	PTSC	86.0	92.0	86.0					

The system was trained by 50% of datasets from Teacher and Student1 and tested for all 100%.

Table XI. Gesture Recognition Results Averaged over 30 Gestures Defined in Table VIII

Testing Data	Descriptor	Average Recognition Accuracy (%)			Testing Data	Descriptor	Average Recognition Accuracy (%)		
		L1	L2	HI			L1	L2	HI
Teacher	PO	77.7	74.3	77.7	Student 1	PO	66.7	66.3	66.7
	PSC	58.0	61.3	57.7		PSC	54.7	56.0	54.7
	PT	96.0	79.3	**96.0**		PT	88.3	73.3	**88.3**
	PTSC	83.0	84.3	83.3		PTSC	79.7	83.0	76.7

These include the reversal of gestures. The system was trained by 50% of datasets and tested for all 100%.

as 92%. This shows the generalization capability of the trained system for recognition of the unseen data.

Next, we used two sets of gestures, Set I and Set II, described in Table IX for the experiment. This database contains 30 gestures, where each gesture G_{ij} has its corresponding reversal G_{ji}. Gesture G_{12} is described by the movement from the first position to the second position, whereas G_{21} represents the movement from the second position to the first position. In this case, the POs of G_{12} and G_{21} may be similar, and thus, they may be incapable of discriminating the two gestures for recognition. The PTs, on the other hand, may preserve the direction of the movement within the gestures, and they may be employed for discrimination of the reversals. This is confirmed by the results shown in Table XI. The proposed system was trained by a (50%:100%) ratio of training samples. Both Teacher and Student1 instances were randomly selected for the training set. It can be observed from the result that the gesture template obtained by PT outperforms other indexing methods discussed. The recognition rate averaged over 30 gesture classes can be attached at 96%. However, the system has a lower performance at about 88% for recognition of the Student dataset. This may be because the dance sequences performed by the student may be inconsistent as compared to the teacher.

6.5. Online Recognition of Continuous Gestures

6.4.1. Progressive Versus Metronome Posture Sampling. In order to assess the utility of the online approach to recognizing and segmenting continuous gestures, the use of histogram intersection directly on a sample of postures (at time t) is initially explored. In this test, we consider two approaches to sampling postures online. In the first, the input sequence is continuously aggregated and converted to a progressive histogram that is matched against the templates for each gesture using the HI metric. Figure 11(a) shows the trace of HI versus sample number for the teacher (performing D1 from Table III). Figure 11(b) shows the same dance D1 for the student. In both cases,

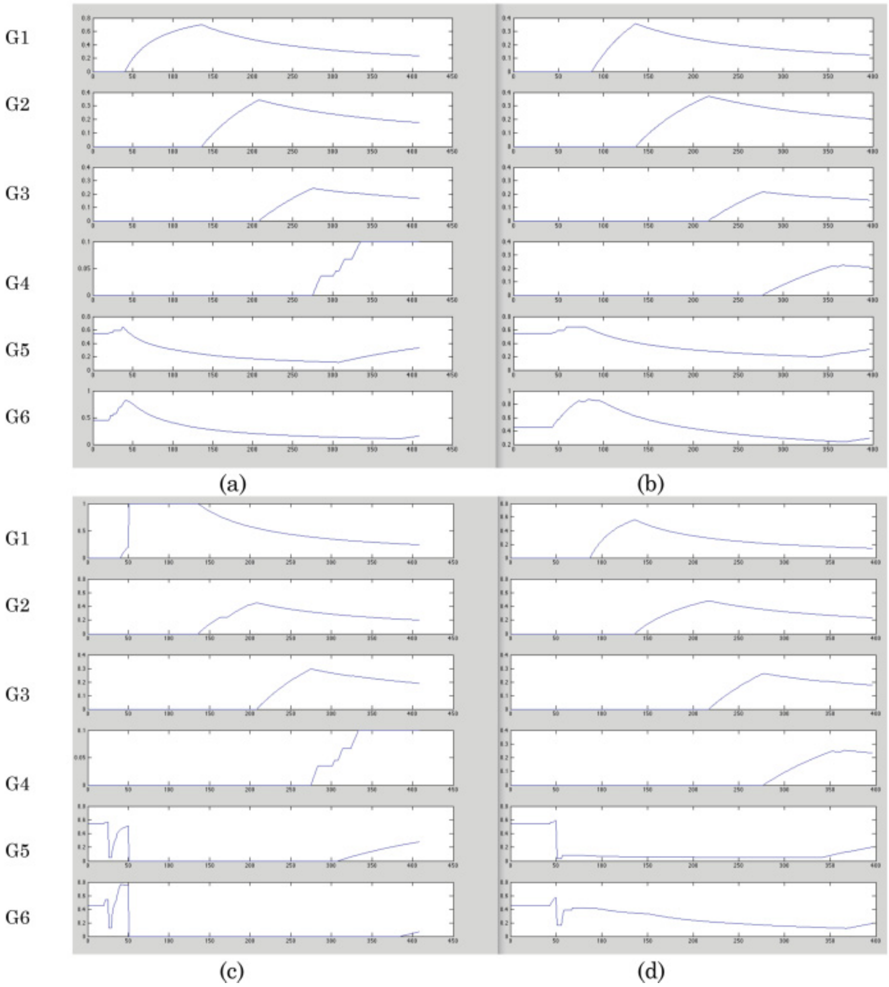


Fig. 11. Continuous dance movement recognition using histogram intersection directly, computed from (1) *progressive* histogram (top): (a) Teacher, (b) Student; and (2) *metronomic* histogram (bottom): (c) Teacher, (d) Student.

even though the set of postures is accumulated from the beginning (and no postures are dropped), there are clear increases in the HI similarity: for example, at frame 50, there is a peak for G6, and at frame 140, G1 peaks, followed by G2, G3, G4, and G5. This is true for both Teacher and Student datasets and corresponds to the expected sequence of gestures (see Table III). The problem is that the degree of similarity is not high, and it becomes difficult to choose a threshold that can work across gestures. In the second test, we consider the fact that the gestures are performed with the guidance of a metronome, which ticks every 50 frames. The metronome is a simplified surrogate for the beat or rhythm that may be associated with music accompanying the dance sequence. Given that the dancer attempts to synchronize with this rhythm, we aggregate postures over the period between metronomic beats. This metronomic histogram results in the set of HI traces in Figures 11(c) and 11(d), each relating to the teacher and student's performance of the dance D1. The result is a somewhat similar pattern

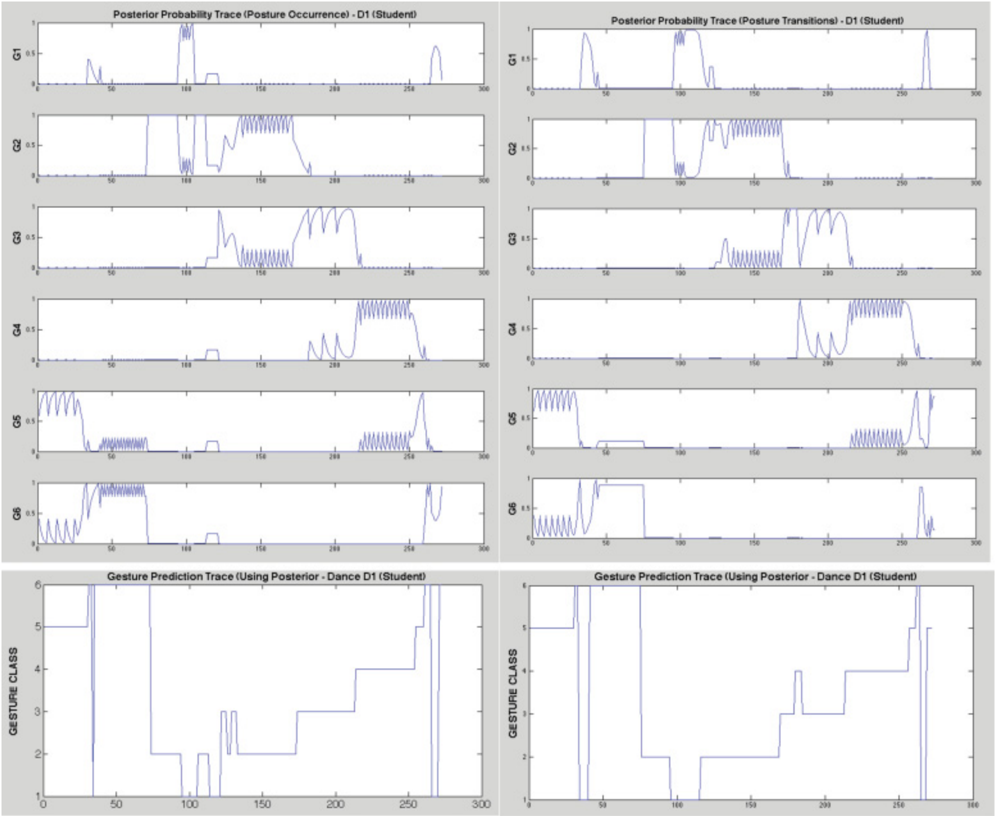


Fig. 12. Online recognition of Teacher (dance D1 gestures). Left top: posterior traces based on posture occurrence. Bottom left: class prediction trace for posture occurrence. Right top: posterior traces based on posture transitions. Bottom right: class prediction trace for posture transitions.

of peaks reflecting the presence of each gesture, with some boosted similarity; however, there is still no clear way to set a detection threshold or condition that can work for all gestures. To this end, we employ the Bayesian framework outlined in Algorithm 2, which will be evaluated in the next section.

6.4.2. Bayesian Recognition Using Histogram Intersection. In the final set of recognition experiments, we evaluate the performance of the proposed Bayesian framework outlined in Section 4.3. In the first test case, the dance D1 is considered, and the online recognition is applied for both the teacher and student, using the posture occurrence and posture transition descriptors, respectively. The posterior probability is captured as a trace (for each gesture class) over the duration of the dance sequence. Results for the teacher sequence are shown in Figure 12, while results for the student are shown in Figure 13.

The results for the teacher show that, for both descriptors, the posterior appears to be quite robust in estimating and switching between gestures. The maximum posterior is selected as the prediction of the gesture class at each time sample in the sequence (shown in Figure 12, bottom left and right). The prediction has been able to extract and segment, in an online manner, the duration of each gesture in the sequence: G6, G1, G2, G3, G4, G5, with some minor noise at the beginning and end of the dance. According to

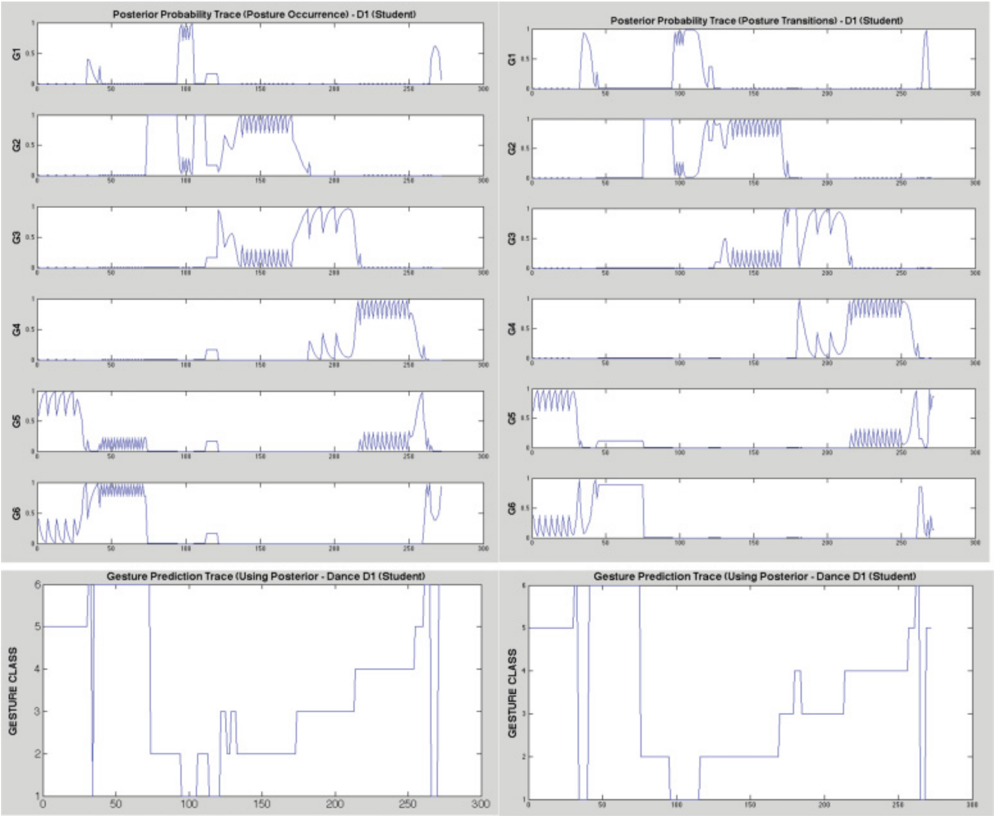


Fig. 13. Online recognition of Student (dance D1 gestures). Left top: posterior traces based on posture occurrence. Bottom left: class prediction trace for posture occurrence. Right top: posterior traces based on posture transitions. Bottom right: class prediction trace for posture transitions.

this result, the system can accurately recognize the teacher's dance gestures from the continuous sequence D1 with 98.6% accuracy (calculated as a percentage of incorrectly detected posture samples over frames 50 to 260, Figure 12). It is apparent that there should be a class to capture derelict cases of postures other than the learned set; otherwise, the posterior will attempt to lock onto the best representation for the input (e.g., G5 at the beginning of the sequence).

The result for the student's performance is also quite satisfactory, as the people performing the movements are different from the teacher and, more so, their ability to repeat the correct movement is somewhat limited. Regardless, with some minor noise, the selection of gesture class appears to follow the actual sequence (i.e., recognition accuracy of 84.3% to 89%, also calculated as a percentage of incorrectly detected posture samples over frames 50 to 260, Figure 13). When confusion does occur, nearby postures are selected for a relatively brief period before switching back to the correct gesture.

One can see that it should also be possible to augment this approach by further employing the metronome idea; one might sample the posterior only at set beats in the rhythm of the dance. As can be seen from the gesture prediction traces in Figure 13, sampling the posteriors at metronomic locations (every 50 frames in this case) would again result in a quite smooth and robust extraction of the correct gesture sequence. It

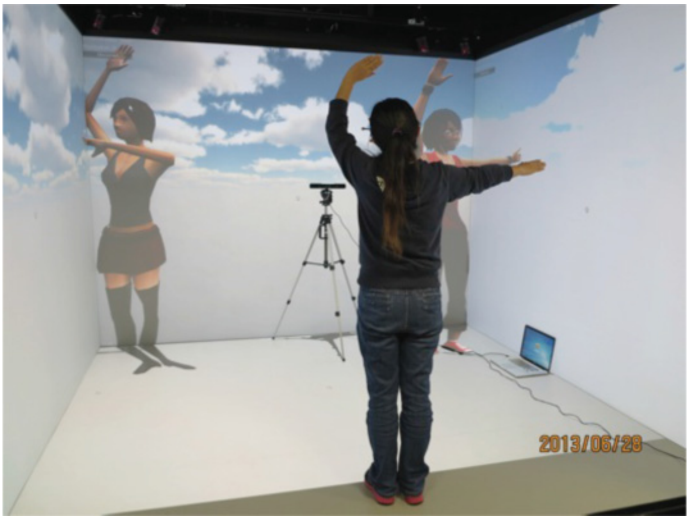


Fig. 14. Illustration of side-by-side feedback.

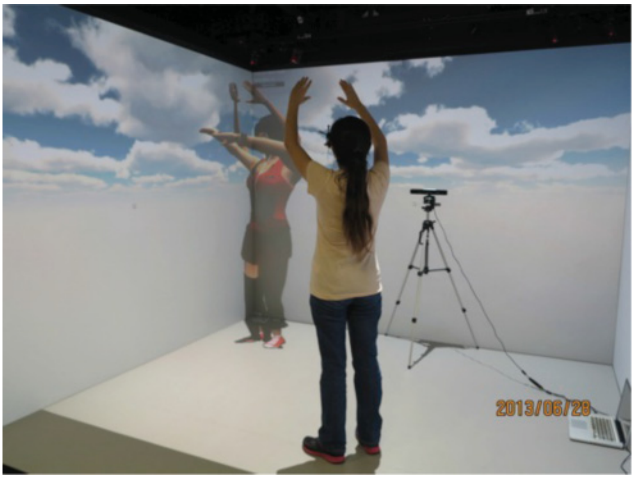


Fig. 15. Illustration of overlay feedback.

is also observed that, with respect to gesture selection, the posture transition provides an improved result over the posture occurrence.

6.6. Results on Student Assessment

Figures 14 to 17 show some pictures of the proposed system for dance training with the student. These include the side-by-side feedback (Figure 14), overlay (Figure 15), and scoring feedback (Figures 16 and 17). In each case, the student wears stereo glasses with optical markers to observe her performance, which allows visualization in 3D.

From the experimental data explained in Section 6.1, we obtained the best teacher dance data and used them as templates for each gesture. The experiments here were aimed at comparing the student's dance performance to the teacher templates after the recognition stage. Figure 18(a) shows the plot of the summation scores computed

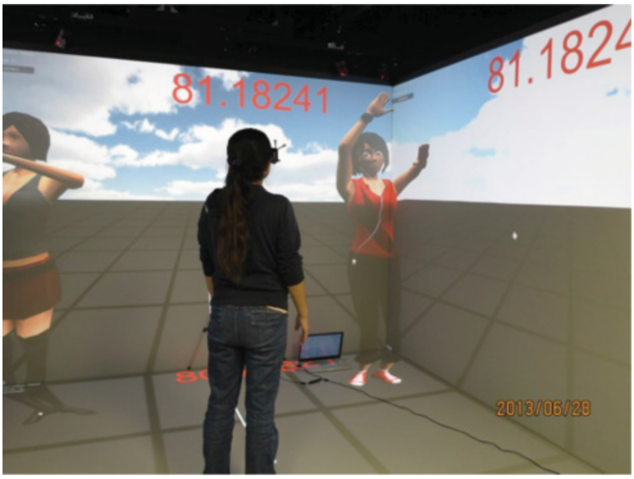


Fig. 16. Illustration of the feedback with overall score.

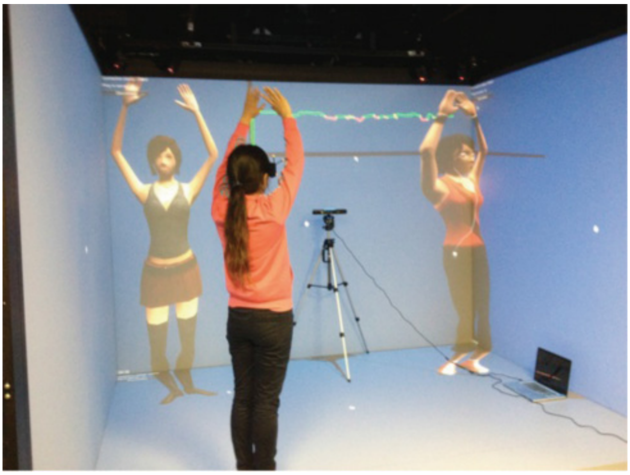
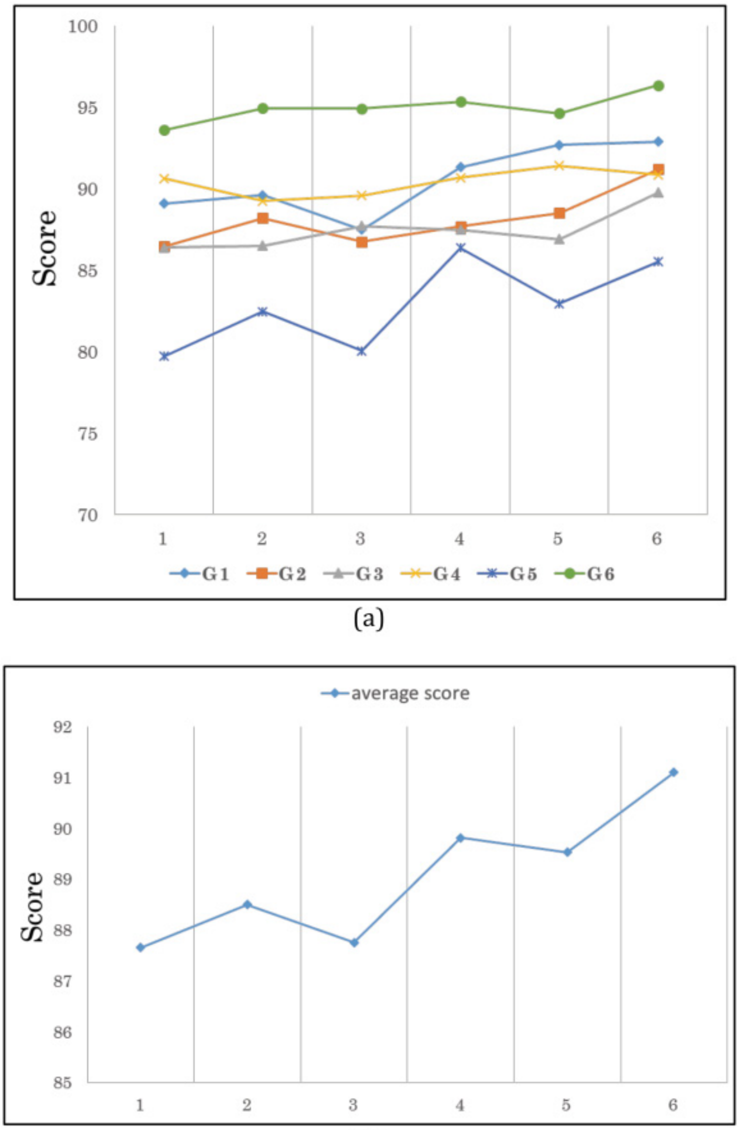


Fig. 17. Illustration of the feedback with score curve.

by Equation (12), describing how the student performs compared to the teacher for each gesture. It can be observed from the figure that students perform well for the sixth repetition at a score of about 95, and the lowest score was observed for student performance on the first time. Figure 17(b) shows the average score for all gestures at six time repetitions of student dances. It is observed that as the student repeated her dance and learned feedback from the system, her dance gestures were close to the teacher.

At this point, the student performance shows only modest improvement. We note, however, that our test routines involve very simple gestures—indeed, the most simple that we use. We made this decision on the basis that we confronted significant technological and aesthetic challenges. The technological challenges involved developing for Kinect, Unity, and Ryerson's Virtual Reality facility. The aesthetic challenges, which are of greater magnitude, involve identifying salient features of these dance gestures to allow us to focus on those divergences between the teacher's and the student's



(b)

Fig. 18. Illustration of the performance of the student's performance compared to the teacher in terms of average score: (a) shows the score when the student performs each gesture six times, and (b) shows the average of results in (a).

performance that are most relevant. Anybody, even people completely lacking in dance training, would be able to reproduce the teacher's performance with a high degree of accuracy from the beginning. In this context, the amount of improvement we would expect is low. The fact that we have observed a detectable, albeit slight, improvement is exactly what we would expect. We therefore take this as evidence that the system is responding well.

We look forward to using more difficult choreographic routines as we develop the system further. We expect, then, that the initial divergence between the student's and the teacher's performance will be greater and that the amount of convergence detected in repeated attempts will increase.

7. SYSTEM LIMITATIONS AND FUTURE WORK

With regard to system architecture, there are several elements that need improvement (work we intend to undertake). The first concerns noise introduced by the Kinect sensor. In our experiment, we tested the system for the recognition of gestures composed of the six positions of basic ballet dance (Figure 8). When we enlarge the set of postures (forms cut in space) beyond that rudimentary set, it becomes important to capture the whole skeleton correctly. In the current version of Kinect, it is required that the dancer face the Kinect; thus, postures that involve bending backward or the occlusion of particular joints are not correctly captured by the system. As a result, the current system has difficulty capturing some balletic movements such as *Pirouette en dehors,* which is a turning movement in which the dancer spins on the spot while standing on one leg with the heel raised. In our experiment, we also observed that Kinect sometimes detected the leg joints inaccurately. It is difficult to capture some dance movements that concern the forms the legs cut in space, movements such as *Grand plié*, *Battement devant, Temps levé*, *Glissade dessus*, and *Grand jeté elancé en avants*. The noise from Kinect affects our recognition and assessment system in two ways. First, as concerns the recognition stage, the resulting SSOM trajectories of dances in the same gesture classes (as shown in Figures 9 and 10) will be slightly different from other samples in the same classes. As a result, the templates of the noisy trajectories result in lower accuracy of performance/gesture recognition. Second, for the visualization and assessment of the dance performance, the similarity score between the skeleton data of the teacher and student may be degraded by the high level of noise from the sensor.

In addition to sensor limitations, it is important to consider the fact that the dataset used in this work is limited in its diversity. In dealing with larger-scale data, it will be important to consider the possibility of incorporating a number of different teachers into the training set. In this sense, the approach taken for training gesture templates using the SSOM will be unchanged, and it is expected that posture transitions and gesture segments that commonly occur will be captured and emphasized—that is, variability between teachers will be naturally de-emphasized by the system. Testing on a larger and more diverse set of students (from a broad range of body types and skill levels) will need to be conducted. In addition, training the SSOM on a full range of detectable ballet postures is also necessary to enable a more complete spectrum of gesture sequences. This also warrants in-depth analysis of SSOM sizing in relation to the number of different postures expected.

Another element that needs further work is the gesture indexing method. In the current implementation, the bag-of-words approach is used to measure the statistics of the coding labels of the SSOM codebook. We have studied only the posture occurrence and posture transition as well as their associations. We have not fully exploited the trajectory of the gestures encoded on the SSOM. It is evident that the transitions from posture to posture (or from one form in space to next) preserve more temporal information about the meter of dance sequence than the postures (forms in space) do themselves, and, consequently, including reversal gestures in the dataset results in higher accuracy (this is discussed in Section 6.4). In order to fully exploit the gesture trajectory on the SSOM, a suitable method for statistical analysis of sequential data such as a hidden Markov model (HMM) is necessary. This may increase the recognition accuracy.

The final element that needs improvement is the 3D visualization in the CAVE. In the current implementation, the visual feedback is provided by the overlay and the side-by-side feedback. Even though we provided a side-by-side feedback mode, the feedback mainly made use of the front projection wall and not much use of the two side walls. This is because the user needs to stand at a distance from the Kinect in order for his or her whole body to be detected. In this case, the two side walls are not fully utilized. We suggest in a future work to make use of the two side walls. For instance, during the visualization of the dance performance, the user can be asked to step forward inside the CAVE and then the teacher's dance motion can be rendered in the front projection as well as in each of the side projections, so that the student can look at the front projection for the front view and the side projections for the left- and right-side views. Then, during the evaluation of the student's performance, he or she can be asked to step back for the Kinect to work properly.

In addition to working on these features, we will extend the complexity and scope of ballet movements and gestures and provide functionality for the online annotation of a user's dance movements as he or she works to interactively construct and review new choreographies.

8. CONCLUSIONS

A novel framework and implementation is presented for the real-time capture, assessment, and visualization of ballet dance movements performed by a student in an instructional, virtual reality (VR) setting. Using both joint positional features and a proposed dance feature (based on angles of joints relative to the dancer's upper and lower torso), a spherical self-organizing map is trained to quantize over the space of postures exhibited in typical ballet formations. Projections of posture sequences onto this space are used to form gesture *trajectories*, used to template a library of predetermined dance movements to be used as an instructional set. Four different histogram models are considered in describing a gesture trajectory specific to a given gesture class (posture occurrence, posture transitions, and sparse codes relating to posture occurrence and transition, respectively).

Recognition performance was evaluated on a database of isolated gesture recordings made by both the teacher and student using three different matching techniques (L1 norm, L2 norm, and histogram intersection). Overall, both features were very effective for recognition, with average recognition rates in the range of 90.5% to 99.5%, with the dance feature showing improved robustness (discounting some errors introduced by derelict/noisy recordings in gestures G4 and G5). The incorporation of posture transitions as a descriptor shows a marked boost in recognition performance (across all matching metrics used) and can be attributed to its detection of temporal ordering of postures. The *bag-of-segments* approach to all four descriptors offers flexibility and generalization across instances of movement recorded from a candidate user: recognition for which, due to the natural variation of the human when repeating movements and the sensor noise introduced by the Kinect, can be a challenging task.

The recognition evaluation was extended to the online case, where a dance composed of continuous gestures is segmented online using a Bayesian formulation of the recognizer (using the histogram intersection metric for computing likelihoods over aggregated postural sequences). This formulation shows much promise (in particular when applied to templates employing descriptors based on posture transition), effectively delineating a student's dance movement into constituent gestural units.

A visualization subsystem compares the detected gestural units against a library of teacher-based gestures and presents immersive visual feedback to the student, thereby quantifying his or her performance. The feedback offers two visual modes for comparing the student's performance of movements with the teacher's and an overall score component to quantify the training session. The virtual environment afforded by the CAVE infrastructure enables the student to experience his or her performance and evaluate it in the same spatial context in which it was performed. This provides unique insight and suggestion for how to adjust and improve enacted dance movements during an interactive training session.

ACKNOWLEDGMENTS

The authors would like to thank Ziyang Zhang, Lei Gao, and Qinxin Deng (Ryerson Multimedia Lab – RML) for their assistance with data collection and CAVE-related testing. In addition, the authors would like to thank Mr. Jordan Sparks for his ongoing contributions to character modeling and animation.

REFERENCES

D. Alexiadis, P. Daras, P. Kelly, N. E. O'Connor, T. Boubekeur, and M. B. Moussa. 2011. Evaluating a dancer's performance using Kinect–based skeleton tracking. In *ACM Multimedia*. 659–662.

O. Arikan and E. Forsyth. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics* 21, 3 (2002), 483–490.

C. W. Armstrong and S. J. Hoffman. 1979. Effects of teaching experience, knowledge of performer competence, and knowledge of performance outcome on performance error identification. *Research Quarterly* 50, 3 (1979), 318–327.

J. Barbic, A. Safonova, J. Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. 2004. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*. 185–194.

M. A. Barnes, D. Krasnow, S. J. Tupling, and M. Thomas. 2000. Knee rotation in classical dancers during the grand plié. *Medical Problems of Performing Artists* 15, 4 (2000), 140–147.

D. A. Becker and A. Pentland. 1996. Using a virtual environment to teach cancer patients T'ai Chi, relaxation, and self-imagery. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*.

M. Bertucco and P. Cesari. 2010. Does movement planning follow Fitts' law? Scaling anticipatory postural adjustments with movement speed and accuracy. *Neuroscience* 171, 1 (2010), 205–213.

A. F. Bobick and J. W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (2001), 257–267.

D. Brennan and M. M. Van Hulle. 2007. Comparison of flat SOM with spherical SOM: A case study. In *The Self-Organizing Maps and the Development – From Medicine and Biology to the Sociological Field.* 31–41. Springer, Tokyo, Japan, (2007).

S. Bronner and S. Ojofeitimi. 2006. Gender and limb differences in healthy elite dancers: Passé kinematics. *Journal of Motor Behavior* 38, 1 (2006), 71–79.

S. Bronner and S. Ojofeitimi. 2011. Pelvis and hip three-dimensional kinematics in grand battement movements. *Journal of Dance Medicine and Science* 15, 1 (2011), 23–30.

J. C. Chan, H. Leung, J. K. Tang, and T. Komura. 2011. A virtual reality dance training system using motion capture technology. *IEEE Transactions on Learning Technologies* 4, 2 (2011), 187–195.

P. T. Chua, R. Crivella, B. Daly, N. Hu, R. Schaaf, T. Ventura, J. Camill, J. Hodgins, and R. Pausch. 2003. Tai chi: Training for physical tasks in virtual environments. In *Proceedings of IEEE Virtual Reality*. 87–94.

A. Clay, N. Couture, and L. Nigay. 2009. Towards an architecture model for emotion recognition in interactive systems: Application to a ballet dance show. In *Proceedings of the World Conference on Innovative Virtual Reality*. 19–24.

A. Couillandres, P. Lewton-Brain, and P. Portero. 2008. Exploring the effects of kinesiological awareness and mental imagery on movement intention in the performance of demi-plie. *Journal of Dance Medicine and Science* 12, 3 (2008), 91–98.

J. L. Deckert, S. M. Barry, and T. M. Welsh. 2007. Analysis of pelvic alignment in university ballet majors. *Journal of Dance Medicine and Science* 11, 4 (2007), 110–117.

L. Deng, H. Leung, N. Gu, and Y. Yang. 2011. Real time mocap dance recognition for an interactive dancing game. *Computer Animation and Virtual Worlds* 22 (2011), 229–237.

K. A. Ericsson, R. T. Krampe, and C. Tesch-Roemer. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological Review* 100, 3 (1993), 363–406.

N. Gamboian, S. J. Chatfield, and M. H. Woollacott. 2000. Further effects of somatic training on pelvic tilt and lumbar lordosis alignment during quiet stance and dynamic dance movement. *Journal of Dance Medicine and Science* 4, 3 (2000), 90–98.

E. Golomer, R. M. Gravenhorst, and Y. Toussaint. 2009. Influence of vision and motor imagery styles on equilibrium control during whole-body rotations. *Somatosensory and Motor Research* 26, 4 (2009), 105–110.

A. O. Gonsales and M. Kyan. 2012. Trajectory analysis on spherical self-organizing maps with application to gesture recognition. In *Proceedings of the 9th International Workshop (WSOM'12)*. 125–134.

K. Hachimura, H. Kato, and H. Tamura. 2004. A prototype dance training support system with motion capture and mixed reality technologies. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*. 217–222.

E. Ho, J. Chan, T. Komura, and H. Leung. 2013. Interactive partner control in close interactions for real-time applications. *ACM Transactions on Multimedia Computing, Communications, and Applications* 9, 3 (2013), 21.

K. M. Holt, T. M. Welsh, and J. Speights. 2011. A within-subject analysis of the effects of remote cueing on pelvic alignment in dancers. *Journal of Dance Medicine and Science* 15, 1 (2011), 15–22.

A. Imura, Y. Iino, and T. Kojima. 2008. Biomechanics of the continuity and speed change during one revolution of the fouette turn. *Human Movement Science* 27, 6 (2008), 903–913.

A. Imura, Y. Iino, and T. Kojima. 2010. Kinematic and kinetic analysis of the fouette turn in classical ballet. *Journal of Applied Biomechanics* 26, 4 (2010), 484–492.

G. Kassing and M. J. Danielle. 1998. *Teaching Beginning Ballet Technique*. Human Kinetics.

E. Kavakli, S. Bakogianni, A. Damianakis, M. Loumou, and D. Tsatsos. 2004. Traditional dance and e-learning: The WEBDANCE learning environment. In *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics*. 272–281.

M. Kawashima, A. Shimada, and R.-I. Taniguchi. 2009. Early recognition of gesture patterns using sparse code of self-organising map. In *Advances in Self-Organizing Maps*. Springer, Berlin, 116–123.

E. Keogh. 2002. Exact indexing of dynamic time warping. In *Proceedings of the International Conference on Very Large Data Bases*. 406–417.

E. Keogh, T. Palpanas, V. B. Zordan, E. Gunopulos, and M. Cardle. 2004. Indexing large human-motion databases. In *Proceedings of the International Conference on Very Large Data Bases*. 780–791.

T. Komura, B. Lam, R. W. H. Lau, and H. Leung. 2006. e-Learning martial arts. *Lecture Notes in Computer Science* 4181 (2006), 239–248.

D. H. Krasnow and S. J. Chatfield. 1997. Imagery and conditioning practices for dancers. *Dance Research Journal* 29, 1 (1997), 43–64.

K. Kulig, A. L. Fietzer, and J. M. Popovich. 2011. Ground reaction forces and knee mechanics in the weight acceptance phase of a dance leap take-off and landing. *Journal of Sports Sciences* 29, 2 (2011), 125–131.

C. F. Lin, F. C. Su, and H. W. Wu. 2005. Ankle biomechanics of ballet dancers in relevé en pointé dance. *Research in Sports Medicine* 13, 1 (2005), 23–35.

F. Lv, R. Nevatia, and M. W. Lee. 2005. 3D human action recognition using spatio-temporal motion templates. In *Computer Vision in Human-Computer Interaction*. Springer, Berlin, 120–130.

N. Masso, A. Germain, F. Rey, L. Costa, D. Romero, and S. Guitart. 2004. Study of muscle activity during releve in first position and sixth positions. *Journal of Dance Medicine and Science* 8, 4 (2004), 101–107.

L. Mayers, S. Bronner, S. Agraharasamakulam, and S. Ojofeitimi. 2010. Lower extremity kinetics in tap dance. *Journal of Dance Medicine and Science* 14, 1 (2010), 3–10.

J. D. McAuley and M. R. Jones. 2003. Modeling effects of rhythmic context on perceived duration: A comparison of interval and entrainment approaches to short-interval timing. *Journal of Experimental Psychology: Human Perception and Performance* 29 (2003), 1102–1125.

M. Naemura and M. Suzuki. 2006. A method for estimating dance action based on motion analysis. In *Computer Vision and Graphics*. Springer Netherlands, 695–702.

J. G. Noverre. 1760. *Letters on Dancing and Ballet*, translated by Cyril W. Beaumont. London 1830, Reproduced, New York Dance Horizons.

M. Raptis, D. Kirovski, and H. Hoppe. 2011. Real-time classification of dance gestures from skeleton animation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 147–156.

A. P. Sangole and A. Leontitis. 2006. Spherical self-organizing feature map: An introductory review. *International Journal of Bifurcation and Chaos* 16, 11 (2006), 3195–3206.

C. Schuldt, I. Laptev, and B. Caputo. 2004. Recognizing human actions: A local SVM approach. In *Proceedings of the IEEE International Conference on Pattern Recognition*. 32–36.

G. Shan. 2005. Comparison of repetitive movements between ballet dancers and martial artists: Risk assessment of muscle overuse injuries and prevention strategies. *Research in Sports Medicine* 13, 1 (2005), 63–76.

J. M. Shippen and B. May. 2010. Calculation of muscle loading and joint contact forces during the rock step in Irish dance. *Journal of Dance Medicine and Science* 14, 1 (2010), 11–18.

R. W. Simmons. 2005. Sensory organization determinants of postural stability in trained ballet dancers. *International Journal of Neuroscience* 115, 1 (2005), 87–97.

J. K. T. Tang, H. Leung, T. Komura, and H. P. H. Shum. 2008. Emulating human perception of motion similarity. *Computer Animation and Virtual Worlds* 19, 3–4 (2008), 211–221.

M. Uejou, H.-H. Huang, J.-H. Lee, and K. Kawagoe. 2011. Toward a conversational virtual instructor of ballroom dance. In *Intelligent Virtual Agents*. Springer, Berlin, 477–478.

R. E. Ward. 2012. *Biomechanical Perspectives on Classical Ballet Technique and Implications for Teaching Practice*. PhD Thesis, University of New South Wales.

M. Wilson, J.-H. Ryu, and Y.-H. Kwon. 2007. Contribution of the pelvis to gesture leg range of motion in a complex ballet movement grand rond de jambe en l'air en dehors. *Journal of Dance Medicine and Science* 11, 4 (2007), 118–123.

U. Yang and G. J. Kim. 2002. Implementation and evaluation of "just follow me": An immersive, VR-based, motion-training system. *Presence: Teleoperators and Virtual Environments* 11, 3 (2002), 304–323.

Y. Yang, H. Leung, L. Yue, and L. Deng. 2012. Automatic dance lesson generation. *IEEE Transactions on Learning Technologies* 3, 5 (2012), 191–198.

Y. Yang, H. Leung, L. Yue, and L. Deng. 2013. Generating a two-phase lesson for guiding beginners to learn basic dance movements. *Computers and Education* 61, 1 (2013), 1–20.

Received July 2013; revised November 2013; accepted January 2014