An Advanced Computational Intelligence System for Training of Ballet Dance in a Cave Virtual Reality Environment

G. Sun[(1)], P. Muneesawang[(2)], M. Kyan[(3)], H. Li[(1)], L. Zhong[(4)], N. Dong[(3)], B. Elder[(3)], L. Guan[(3)]*
(1) Communication University of China, (2) Naresuan University (3) Ryerson University
(4) Guangdong University of Technology
e-mail: paisarnmu@nu.ac.th

***Abstract*—This paper presents a computer-based system for assessment and training of ballet dance in a CAVE virtual reality environment. The system utilizes Kinect sensor to capture student's dance and extracts features from skeleton joints. This system depends on a structured posture space, which comprises a set of dance elements that represent key moments—"postures", that typically will be so briefly held as to experience as a fleeting moment in a flux —in the dance movements whose performance we are attempting to assess. The recording captured from the Kinect allows the parsing of dance movement into a structured posture space using the spherical self-organizing map (SSOM). From this, a unique descriptor can be obtained by following gesture trajectories through posture space on the SSOM, which appropriately reflects the subtleties of ballet dance movements. Consequently, the system can recognize the category of movement the student is attempting, and this allows us make a quantitative assessment of individual movements. Based on the experimental results, the proposed system appears to be very effective for recognition and offering generalization across instances of movement. Thus, it is possible for the construction of assessment and visualization of ballet dance movements performed by the student in an instructional, virtual reality setting.**

Keywords-dance traning system, dance assesment, CAVE virtual reality environment, gesture recognition, spherical-self-organizing map

I. INTRODUCTION

Classical ballet is based on distinct aesthetic ideals and disciplined style of dance. Its thoroughly idealized movements require an extreme precision, achievable only by those whose have developed muscular strength, a strong core stability, and an extraordinary learned posture that enables movements that untrained people cannot perform. This strength and flexibility are acquired through long practice, some of which involves repetition of established movement patterns. Ballet dance training, therefore, requires an effective assessment method for the precise alignment of postures of trainees. This paper presents a computer-based system to address this issue based on two techniques: automatic dance gesture recognition, and the 3D visual feedback to effectively assess student performance and training.

In traditional training, students learn physical and mental skills by the instructors giving a demonstration which will then be imitated by the students [1]. Instructors supervise their students with feedback, informing how student movement response compared to the ideal template for their particular discipline. The efficacy of this feedback depends largely on the instructor's ability to identify the aspects of the response. At this point, however, ballet dance training relies more on qualitative rather than quantitative assessment. As such, a number of research works based on quantitative methods have appeared that have attempted to develop an objective and systematic means of analysis of ballet dance techniques [2]-[4]. The kinematic and kinetic data, captured by video and 3D motion analysis, are utilized in the investigation of the biomechanical properties of human movement.

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. On this model, the value of the measurements taken (i.e., score), 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 gestures' 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 [5] or the synthesis of dance partners "on the fly" [6].

We take the quantitative dance training system further by adding computational intelligence, to enable the system to recognize the student's dance gesture, assessment, and training in a fully immersive virtual reality system, the CAVE. With an explicit model of a students' gestures, assuming a desired goal, the proposed system uses a trajectory of postures exposed through SSOM learning, to predict the target gestures, given their actions. The dance teaching problem is thus inverted into the problem of predicting the student's gestures. This is followed by an computational assessment of the student's performance and display for the student (and teacher) comparing the performances, which provide the student with visual feedback in CAVE environment, allowing the student to see clearly where his or her performance meets expectations and where it falls short. Figure 1 shows the architecture of the proposed system which includes four components: Kinect motion capture, CAVE, gesture recognition, and gesture database.

The first contribution of this paper is the presentation of the new method for the trajectory analysis on SSOM for gesture recognition. In the proposed gesture recognition algorithm, postures are represented by a particular state of sensor values. MS Kinect uses skeletal tracking of joint positions, which are then mapped onto the SSOM. A gesture can then be represented as a path or trajectory on the map, as

978-1-4799-4311-1/14 $31.00 © 2014 IEEE
DOI 10.1109/ISM.2014.55

traced by projecting a temporal series of postures. The trajectory analysis methods are then applied for the gesture recognition. In comparison to the previous works for gesture recognition in [7]-[9] that implement the self-organizing map (SOM) for quantization of features, the current work adopts SSOM which provides better visualization on the spherical structure [10]. Moreover, for the trajectory analysis, the previous methods have attempted to use sparse codes [7]-[9], and Bag-of-Word (BoW) model [11]. These methods have some limitation of the capability for the temporal information analysis since they consider only the existing of codewords and the frequency of occurrence of the nodes on maps. In order to perform trajectory analysis on SSOM, in this paper, a method for transition analysis on the SSOM trajectory is presented. It is derived from the Markov empirical transition matrix and effective for capturing temporal information of the dance gestures. In comparison to the works in [7]-[9], [11], this paper also addresses the real-time segmentation of dance gesture from the continuous ballet dance using the proposed trajectory analysis method.

The second contribution is the development of a complete VR dance training system that comprise of gesture recognition, gesture segmentation, dance assessment, and VR feedback. 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 [12]-[15]. Under this paradigm, learning ability is entirely based on the virtual representation 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 necessary result in improved performance. In some recent papers [16]-[20], an alternative to this learning paradigm was proposed, in which the student assessment can be performed with rapid feedback using the standard automated protocol. The main activities in this approach consist of analyzing a student's motion against the designed (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 [21], 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 these important requirements that arise in connection with standard methods of teaching element 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. The performance evaluation is provided in the form of 3D visualizations and feedback through a CAVE virtual environment. In an off-line process, the movements of a teacher are represented as gesture trajectories through unsupervised posture space on the SSOM. Four types of templates, are utilized for indexing the gesture trajectories. The system facilitates visual alignment and score-based assessment of individual movements within the context of the dance sequence.

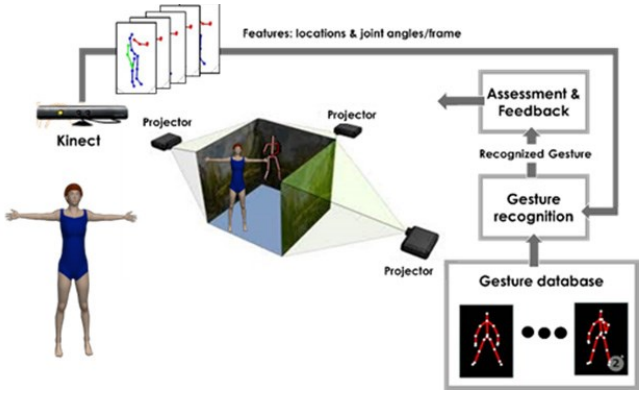


Figure 1. System architecture

II. Gesture Recognition

The gesture recognition module can be described in the following three parts: building a posture space, building gesture templates, and recognition.

A. Building Posture Space

The gesture recognition process begins with automatically parsing the samples, from across the spectrum of expected dance movements, into a discrete set of postures. We adopt the SSOM demonstrated in [22] for this parsing process. Figure 2 shows the structure of SSOM where the postures are projected (quantized) on to the map. The utility of a SSOM 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, i.e. postures are positioned in the map nearby to other postures that are very similar in nature. As a consequence of this topology-preserving mapping, a sequence of expected "posture moments" within an ongoing flow of movements could be expected to trace a comparatively smooth trajectory on the map. It is from this trajectory (sequence of key postures) that we formulate descriptors representing each gesture.

The construction of posture space amounts to training the SSOM using a random set of sample postures from the supervising set of gesture movements. We define the training set $\boldsymbol{X}_t$ as the set of (teacher's) gestures $\boldsymbol{g}_{c,n}$:

$$\boldsymbol{X}_t = \{\boldsymbol{g}_{1,1}, \boldsymbol{g}_{1,2}, \boldsymbol{g}_{1,3}, \dots, \boldsymbol{g}_{c,n}\} \quad (1)$$

$\boldsymbol{g}_{c,n}$ is the n^{th} instance (recording) of the c-th gesture class,

$$\boldsymbol{g}_{c,n} = \{\mathbf{x}_{c,n}^1, \mathbf{x}_{c,n}^2, \dots, \mathbf{x}_{c,n}^t\} \quad (2)$$

where $\mathbf{x}_{c,n}^t \in \mathcal{R}^D$ is the posture feature vector of $\boldsymbol{g}_{c,n}$ at time t, and D is the dimension of the feature vector.

The training phase of the SSOM is summarized in Algorithm 1. Let $\mathbf{w}_{i,j,k}$ be the weight vectors of the $(i,j,k)^{th}$

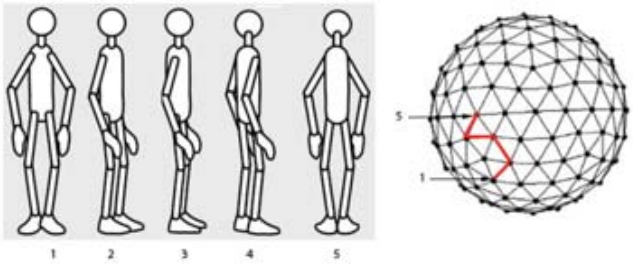


Figure 2. Temporal sequence of postures representing an arbitrary movement. A simple 'gesture' of 5 postures is displayed. The mapping of the gesture forms a trajectory on the SSOM in red.

cluster of the spherical lattice map, where (i, j, k) are the index of the cluster units in the three dimension. Each cluster unit at (i, j, k) has a variable neighbourhood $NE_{i,j,k}$ with a decreasing radius. Giving the input space $\boldsymbol{X}_t$ and the weight vectors $\mathbf{w}_{i,j,k}$, the system carries out the following learning process [22].

Firstly, the input posture vectors $\mathbf{x}^i$ are randomly introduced into the SSOM. For each voxel, the best matching unit (BMU), $\mathbf{w}_{(i,j,k)*}$ is selected. BMU is the nodes which is closest to the input voxel according to the L2 norm similarity measure. *Secondly*, information from $\mathbf{x}^i$ is imparted to $\mathbf{w}_{(i,j,k)*}$ and the weight vectors in this node's immediate neighbourhood. 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. *Thirdly*, these learning steps are repeated. *Finally*, learning is terminated, typically after a fixed number of iterations or changes in node weights become negligible.

B. *Trajectory Analysis using Sparse Code and Bag-of-Word*

Training the individual gesture templates involves projecting a set of labelled 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.

Let $\boldsymbol{g} = \{\mathbf{x}^1, \dots, \mathbf{x}^t, \dots, \mathbf{x}^T\}$ be an input gesture, where $\mathbf{x}^t$ is the posture feature vector of the gesture at time t, and T is the total of postures. Also, let $\mathbf{w}_m, (1 \dots m \dots M)$ denotes the set of weight vectors, where M is the total number of the SSOM nodes. This set of weight vectors is the output of Algorithm 1. The input vectors $\mathbf{x}^t, (1 \dots t \dots T)$ can then be transformed from a series of postures to a series of map units based on their best matching units (BMUs), i.e.,

$$\mathcal{S} = Q(\boldsymbol{g}) = (u_1, \dots, u_t, \dots, u_T), t \in [1, T] \quad (3)$$

$$u_t = \underset{m}{\operatorname{argmin}}(\|\mathbf{x}(t) - \mathbf{w}_m\|) \quad (4)$$

where $Q(\boldsymbol{g})$ is the quantization operation and u_t is the index of the BMU of the input $\mathbf{x}^t$.

ALGORITHM 1. Spherical Self-Organizing Map

input: map configuration
output: weights for all nodes in the map $\mathbf{w}_{(i,j,k)}$

Initialize weights $\mathbf{w}_{(i,j,k)}$ (small random values)
Initialize the number of cycles, N_{cycle}
Initialize the maximum number of epochs, *Max Epochs*
repeat
  Get next input: $\mathbf{x}^i$ = randomly select from training set $\boldsymbol{X}_t$
  Calculate node error: $E^i_{i,j,k} = \varphi(u_{i,j,k}) \sum_{n=1}^{D} (x^i_n - w_{n,i,j,k})^2$
  Select BMU: $(i, j, k)^* = min\{ E^i_{i,j,k} \}$
  Update BMU & neighbors:
    $\mathbf{w}_{(i,j,k)^*}(new) = \mathbf{w}_{(i,j,k)^*}(old) + \alpha[\mathbf{x}^i - \mathbf{w}_{(i,j,k)^*}(old)]$
  where:
    $\alpha = \mu\left(\frac{NE_{(i,j,k)^*}}{NE_{initial}}\right)$ = predefined learning rate
    $NE_{(i,j,k)^*}$ =neighbourhood of BMU
    $NE_{initial}$ = initial neighbourhood size (radius)
    $\varphi(u_{i,j,k})$ = count dependent, non-decreasing function used to prevent cluster under-utilization

  Increment N_{cycle}
until $\boldsymbol{N_{cycle}}$>*Max Epochs*

Given the sequence $\mathcal{S}$ traced on the SSOM, we first consider two types of descriptors which are the sparse code and posture occurrence for gesture indexing.

The sparse code (SC) method has been utilized for structuring the coding labels of the hierarchical SOM [7]-[9]. This method is adopted in the current work, and compared to the newly proposed method. During mapping of posture vectors, the weight vectors $\mathbf{w}_m, (1 \dots m \dots M)$ are labeled as the activated nodes if they are the winning nodes according to Eq. (4). Each node has a state S that it is winner for a gesture element or not, and whole state of the nodes are used as the output, $SC = (S_1, \dots, S_m, \dots, S_M)$. The S_m is defined as follow.

$$S_m = \begin{cases} 1, & \text{if } m = u_t|_{t \in [1,T]} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

S_m is regarded as a sparse code which represents an activated pattern of winner node for a gesture element. The sparse code only represents the existence of a set of postures, and not their frequency of occurrence.

Based on the SC method, a sparse code template (SCT) over the set of gesture instances for the *i*-th class can be computed as the reference template, i.e.,

$$SCT_i = \frac{\sum_{n=1}^{N_i} SC_{i,n}}{\left|\sum_{n=1}^{N_i} SC_{i,n}\right|} \quad (6)$$

where N_i is the number of gesture instances in the *i*-th gesture class, and $SC_{i,n}$is the sparse code of the *n*-th gesture instance belonging to the *i*-th class.

Posture occurrence (PO) is analogous to the popular BOW approach adopted for pattern recognition task [11]. By aggregating the occurrence of postures in a gesture against the indexed set of nodes on the map, a histogram can be formed (over a single gesture, or set of similar gestures), thus forming a template that is used in recognition. In this method, a

histogram is formed for each gesture instance n in the i-th gesture class, $PO_{i,n} = [H(1), \dots, H(m), \dots, H(M)]^t$. The value of the m-th component is calculated by:

$$H(m) = \sum_{t=1}^{T} \delta(u_t - m), m \in \{1, \dots, M\} \quad (7)$$

where δ is the Kronecker delta function, and T is the total number of indices in the sequence of node indices $\mathcal{S}$ discussed in Eq. (3).

A reference template for the i-th gesture class can be formed by summing over the set of $PO_{i,n}$:

$$POT_i = \frac{\sum_{n=1}^{N_i} PO_{i,n}}{\left|\sum_{n=1}^{N_i} PO_{i,n}\right|} \quad (8)$$

where N_i is the number of gesture instances in the i-th gesture class and $PO_{i,n}$ is the posture occurrence vector of the n-th gesture instance belonging to the i-th gesture class.

C. Trajectory Analysis using Posture Transition Model

The SC and PO methods do not consider the temporal arrangement of postures in the map. They only consider the occurrence of map units and the frequency of the individual nodes for indexing. We observe that these methods have made great efforts to maintain the marginal histogram of the SSOM indices (first order statistics). Since the gesture contains postures which have somewhat strong correlation with their neighbour, the adoption of the second order statistics, such as covariance and co-occurrence matrix, are more appropriate for capturing the dependency between the pairs of postures from the SSOM trajectory. Based on this discussion, a feature extraction based on posture transition (PT) is obtained as follows.

Given that u_t is the index of a map unit, the function in Eq. (4) creates $\mathcal{S} = (u_1, \dots, u_t, \dots, u_T)$–the set of indices of map units treated as a set of symbols. The u_t value of subsequent points of a gesture remains the same since subsequent points are generally close in the input data space. As a consequence, equal values of u_t are replaced with a single value which results in the following definition [23]:

$$Tr = \mathcal{N}(\mathcal{S}) = \{u'_1, \dots, u'_w, \dots, u'_W\} : W \leq T, \quad (9)$$

$$u'_i \neq u'_{i-1}, \forall i \in [2, W], \quad (10)$$

where $\mathcal{N}(\cdot)$ is a function that removes consecutive equal u_t values and Tr is the mapped gesture, representing the trajectory on the SSOM. By the arrangement in Eq. (9), the dependencies among neighboring nodes can be conveniently investigated.

The Markov random process is employed to model the trajectory. To capture the dependencies between SSOM nodes in the trajectory, the horizontal Markov empirical transition matrix [24] of the dataset in Tr is calculated. The matrix's element is given by the probability:

$$P_h(u'_{i+1} = n | u'_i = k) = \frac{\sum_{i=1}^{W-1} \delta(u'_i = k, u'_{i+1} = n)}{\sum_{i=1}^{W-1} \delta(u'_i = k)} \quad (11)$$

where u'_i and u'_{i+1} are neighboring node indices pair, W is the size of Tr, and $w, n \in \{1, \dots, M\}$.

$$\delta(x) = \begin{cases} 1, & x = \text{true} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

Based on Eq. (11), the dimension of the transition matrix is $M \times M$ since the SSOM has M nodes. The PT feature vector is formed by arranging the element of this matrix, $P_h, h = 1, \dots, M^2$, into a 1-D template.

In addition, we can also obtain the posture transition sparse codes (PTSC) that are analogous to sparse codes of postures, only, they represent the existence of transitions rather than the frequency of transitions.

D. Isolated Gesture Recognition

In order to match an incoming gesture $\boldsymbol{g}_u$ with a known template (discussed in Section II(B)-(C)), the incoming set of postures is projected onto the SSOM to extract the unknown posture sequence $\mathcal{S}_u$ (according to Eq. (3)). We propose an online probabilistic framework (inspired by the work of Kawashima et al. [7]. We adopt a simple Bayesian framework for progressively estimating an updated posterior probability $P(c|\mathcal{S}_u)$ for each of the $c = 1..C$ gesture classes. The likelihood $P(\mathcal{S}_u|c)$ is computed as the ratio of the number of incidents of the current posture in gesture class c to the total number of different postures in class c. In this work, we reframe the likelihood as a histogram intersection, between a progressively growing sequence $\mathcal{S}_u$ (inclusive of postures from time t_0 to t), which may be described as a histogram of either: PO or PT (discussed in Section II(B)-(C)), versus the corresponding template histograms for each gesture class.

We consider $\mathbf{h}_s$ to be the histogram feature for the current sample at time t, and $\mathbf{h}_c$ to be the histogram template for the class c. We thus define (for time t), the posterior $P_t(c|\mathbf{h}_s)$, likelihood $P_t(\mathbf{h}_s|c)$, and prior $P_t(c)$ probabilities according to the following:

$$P_t(c|\mathbf{h}_s) = \frac{P_t(\mathbf{h}_s|c)P_t(c)}{P_t(\mathbf{h}_s)} = \frac{P_t(\mathbf{h}_s|c)P_t(c)}{\sum P_t(\mathbf{h}_s|c)P_t(c)} \quad (13)$$

$$P_t(\mathbf{h}_s|c) = HI(\mathbf{h}_s, \mathbf{h}_c) \quad (14)$$

$$P_t(c) = \begin{cases} \frac{1}{C}, & if\ t = t_0 \\ \frac{P_{t-1}(c|\mathbf{h}_s) \cdot HI(\mathbf{h}_s, \mathbf{h}_c)}{\sum_C P_{t-1}(c|\mathbf{h}_s) \cdot HI(\mathbf{h}_s, \mathbf{h}_c)}, & otherwise \end{cases} \quad (15)$$

$$HI(\mathbf{h}_s, \mathbf{h}_c) = 1 - \sum_i \min\left[h_{s,i}, h_{c,i}\right] \quad (16)$$

According to equations above, 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 PO or PT. Likelihood's are estimated as histogram intersections, Eq. (14), 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.

As the sequence begins to resemble a gesture from the known set, its posterior probability will grow, and eventually surpass a detection threshold. Upon triggering this threshold, the class c with the maximum posterior probability 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.

III. Dance Assessment and Feedback

An impediment to research on virtual reality is the lack of degree 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 student's eye movements to be tracked. 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.

Once the system identifies the best matched gesture class, the remaining problem is to determine how well the student has performed the phrase compared to the teacher. The feedback methods are shown in Fig. 5, and described as follows:

Side by Side: virtual models play back the most recent performance of the student and the teacher side-by-side. 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. Figure 5(a) shows the side-by-side feedback.

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. Figure 5(b) shows the overlay feedback.

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 a particular time t is calculated as follows:

$$SC[t] = 1 - d[t] \tag{18}$$

$$d[t] = \sum_{i=1}^{19} \left| \frac{\theta_i^s[t] - \theta_i[t]}{\max_i \theta_i[t] - \min_i \theta_i[t]} \right| \tag{19}$$

where $SC[t]$ is the score at time t, $d[t]$ is the relative distance at time t between the student's features and teacher's, $\theta_i[t]$ and $\theta_i^s[t]$ are the i^{th} feature joint angles of the teacher and the student, respectively. Here, we extract a set of 19 features from the 20 3D skeleton matrix from Kinect, as the method discussed in [25]. The score curve obtained by Eq. (18) 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.

IV. Experimental Results

The proposed system outlined in Fig. 1 was implemented. The CAVE has 4 stereoscopic projectors and screens correspondingly. Driven by a graphics cluster of 5 nodes 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 user's eyes are tracked 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 located at the top of the screens. The tracking data is used to determine the content to be display on the screens. We used 3D Unity game engine and Visual C# to implement the feedback engine, and interface with the Kinect sensor. MiddleVR was used to control the graphic in the CAVE.

A. Ballet Dance Gestures

We implemented the system with the database of dance within the six basic positions (i.e., six postures) of ballet dance. Based on these six postures, we define a 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 I shows a matrix describing the definition of all gestures. In the table, giving the six postures P_1-P_6, the gesture G_{ij} is formed as an isolated gesture moving from the *i*-th position to *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, which has a total of gestures of $N = 15$. 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 by the number of membership in Set I and Set II, which is $2 \times N = 30$ gestures.

B. Feature Extraction

The 3D motion capture module was implemented by the Microsoft Kinect, which records frames, each of which contains 20 3D skeleton points to represent a student in the camera's field of view. We denote each joint location as a feature in time series, x_i^t where t is the time index, and x_i is the location of the *i*-th joint in one of the x/y/z planes. The joint locations were normalized by using the hip as the origin, and calculating the location of each joint relative to the hip. Thus, the normalized locations of all 20 joint positions, can be described by: $\boldsymbol{x}^t = \{\hat{x}_i^t, i = 1, \ldots, 60\}$, where $\boldsymbol{x}^t$ is the posture feature vector of a gesture at time t, $\hat{x}_i^t$ is the location of the *i*-th joint normalized by using the hip as the origin. All 20 joints in the 3 dimensions were considered resulting in 60 dimensions.

C. Static Gesture Evaluation

We first used the non-reversal gestures in Set I. Two datasets were constructed: Teacher dataset and Student

datasets. The Teacher dataset and Student dataset were used for both construction and testing of gesture recognition performance. The database includes 15 isolated gestures (i.e., each gesture is recording independent of any sequence of other movement/gesture). The structure of this dataset is summarized in Table II. In order to assess the performance of the SSOM posture space representation, gesture template definitions and matching criteria, the system was trained by (50%:100%) ratio of training samples. From the full set of Teacher gestures and Student gesture, 50% (e.g., 10/20 instances from each gesture) were randomly selected and used to form gesture templates, while all 100% were classified against these templates.

The system employed the SSOM [cf. Algorithm 1] with the following configuration: icosahedron level = 2, map nodes = 162, neighborhood = 4, and epochs = 100. Figure 3 shows a series of mappings of gesture instances (columns) per gesture type (rows).

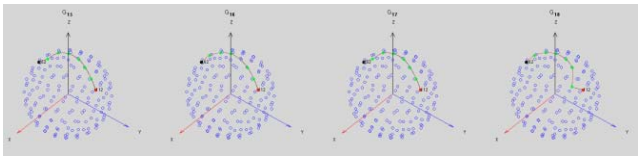
A visualization of the SSOM and associated gesture trajectories shows that even differences in frame length and duration of the gesture (variations of up to 40% difference in frame length) do not appear to impact the consistency with which the gesture maps onto posture space. All gestures appear to trace 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.

Table III shows the performance of the system for recognition of ballet dance performed by two people, Teacher and Student. The system can attain more than 98% recognition rate averaged over 15 classes for recognition of Teacher dataset by using PT template and HI for similarity matching. The PO template also gave similar recognition performance to the PT method. Moreover, the system can recognize dance from Student dataset at 100% accuracy by using the PO template and L2 norm for similarity matching. 92%.

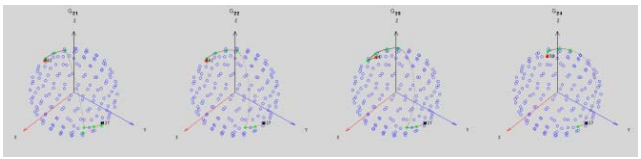
Next, we used two sets of gestures, Set I and Set II described in Table II for the experiment. This database contains 30 gestures, where each gesture G_{ij} has its corresponding reversal G_{ji}. Gesture G_{12} is described by the movement from the 1st position to the 2nd position, whereas G_{21} represents the movement from the 2nd position to the 1st position. In this case, the posture occurrences (POs) of G_{12} and G_{21} may be similar, and thus, they may be incapable for discriminating the two gestures for recognition. The posture transitions (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 IV. 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 student dataset. This may be because the dance sequences performed by the student may be inconsistent, as compared to the teacher.

D. Result for Online Gesture Recognition

In order to assess the online capability of the system to recognize isolated gestures from a continuous dance phrase,



(a) Four instances of gesture G_{12}



(b) Four instances of gesture G_{23}

Figure 3. Gesture Projections: instances of gestures G_{12} and G_{23} (rows top-bottom) respectively. Smooth, local sets of postures show stable, highly repeatable trajectories.

TABLE I. DEFINITION OF THE THIRTY GESTURES. P_i IS THE *i*-TH POSTURE. G_{ij} IS THE GESTURE PERFORMING FROM THE I-TH POSTURE TO THE *j*-TH POSTURE. G_{ij} IS THE REVERSAL OF THE GESTURE G_{ji}.

	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}	-

TABLE II. ISOLATED GESTURE DATABASE

Gesture	**# Instance for each gesture**		**Total instances**
	Teacher	***Student***	
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	300
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	300

the fourth dataset was constructed. In this, recordings were collected for two movement phrases; one for Teacher and one for the Student. The gesture sequence of this continuous phrase is G_{61}, G_{12}, G_{23}, G_{34}, G_{45}, G_{56}. The Teacher's phrase contained 281 frames, while the Student's contained 273 frames. We evaluate the performance of the proposed Bayesian method (discussed in Section II(D)). Our online recognition method was applied to both the Teacher's and Student's performances, using the posture occurrence descriptor. The Posterior probability is captured as a trace (for each gesture class) over the duration of the dance phrase. Results for the Teacher sequence are shown in Fig. 4(a), while results for the Student are shown in Fig. 4(b).

The results for the Teacher shown that, the posterior probability appears to be quite robust in estimating and switching between gestures. The maximum posterior probability is selected as the predicted gesture class for each time sample in the sequence (shown in Fig. 4(a) - bottom).

The prediction has been successful in performing an online extraction and segmentation for the duration of each gesture in the sequence: G_{61}, G_{12}, G_{23}, G_{34}, G_{45}, G_{56}, with some

TABLE III. GESTURE RECOGNITION RESULTS AVERAGED OVER 15 GESTURES DEFINED IN THE UPPER TRIANGLE IN TABLE I.

Testing Data		Average Recognition Accuracy (%)		
		L1[a]	L2 [a]	HI [a]
Teacher	PO	96.7	98.0	96.7
	PSC	79.3	84.0	79.3
	PT	98.7	97.3	**98.7**
	PTSC	87.3	92.7	87.3
Student	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

a. The template matching was performed by three similarity metrics: L1 norm, L2 norm, and histogram intersection (HI).

TABLE IV. GESTURE RECOGNITION RESULTS AVERAGED OVER 30 GESTURES DEFINED IN TABLE I.

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

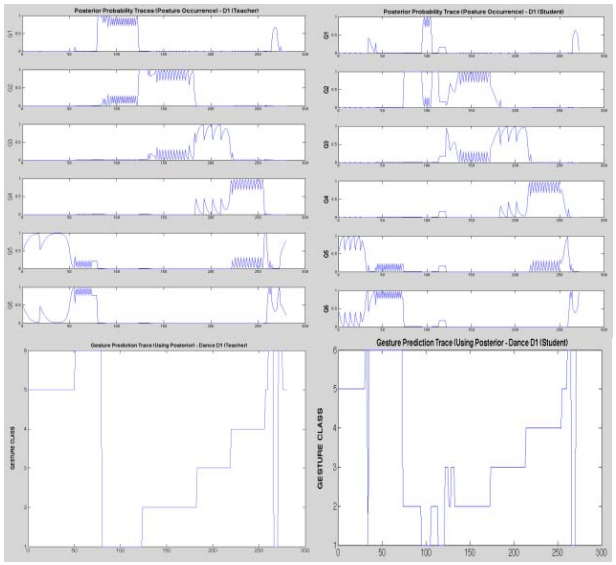
TABLE V. CONTINUOUS GESTURE DATASET

Ballet Dance		**# instances (# frames)**	
Label	Postures (Gesture Sequence)	Teacher	Student
D1	Rest→1st→2nd→3rd→4th→5th→Rest (G_{61}, G_{12}, G_{23}, G_{34}, G_{45}, G_{56})	1 (281)	1 (273)

minor noise at the beginning and end of the performance. The result for the Student's performance seems quite satisfactory, as the people performing the movements are physically (and kinesiologically) different than the Teacher, so they cannot be expected to repeat identically a given gesture. Regardless, with some minor noise, the selection of gesture class appears to follow the actual sequence.

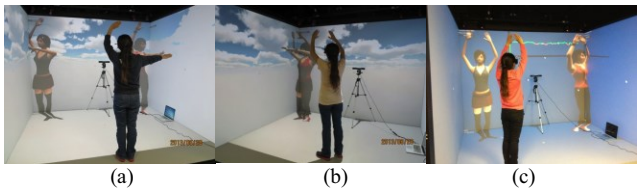
E. Result of Student Assessment

Lastly, we obtain the results of the student assessment. Figure 5 shows some pictures of the proposed system for dance training with the student. These include the side-by-side feedback (Fig. 5(a)), overlay (Fig. 5(b)), and scoring feedback (Fig. 5(c)). In each case student wears stereo glasses with optical markers to observe her performance, which allows visualization in 3D. From the experimental data discussed previously, we obtained the best teacher dance data and used them as templates for each gesture. The student dance was recognized and compared to the teacher dance. This time, the student performed and received feedback. The training was repeated 6 times. Figure 6 shows the average score (computed by summation of scores in Eq. (18)) for all gestures and 6 repetitions of student dances, describing how a student performs, compared to the teacher. It can be observed that the



(a) (b)

Figure 4. Online recognition of continuous dances of Teacher (a) and Student (b); top: Posterior traces based on posture occurrence; bottom: class prediction trace for posture occurrence.



(a) (b) (c)

Figure 5. Illustration of (a) side-by-side feedback, (b) overlay feedback, and (c) the feedback of score curve.

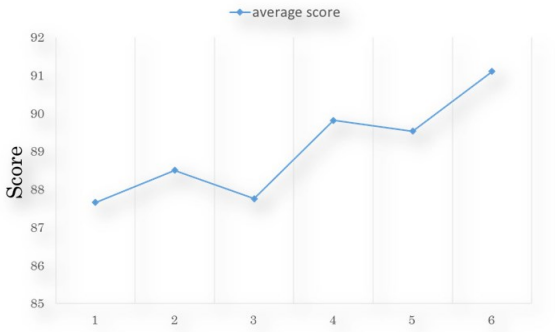


Figure 6. Illustration of the student's performance compared to teacher in terms of score averaged over all gestures in a continue dance sequence, for 6 time repetitions of student dances.

student performed well at the 6th repetition, with an average score of 91, and the lowest score and that the lowest score measured was for the first performance. It can be observed that as the student repeated her dance and received feedback from the system, her dance gestures approximated more closely those of the teacher.

F. Limitations

In our experiment, we tested the system for the recognition of gestures composing of the 6 positions of basic ballet dance. When we enlarge the set of postures 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 to the Kinect and postures that involve bending backwards is not correctly captured by the system. We also observed that the Kinect sometimes detected the leg joints inaccurately.

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/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.

V. CONCLUSION

A new 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 joint positional features, 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. Two different histogram models are considered in describing a gesture trajectory specific to a given gesture class (posture occurrence and posture transitions). The histogram approach to all two 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. This formulation shows much promise, effectively delineating a student's dance movement into constituent gestural units.

REFERENCES

[1] C. W. Armstrong, and S. J. Hoffman, "Effects of teaching experience, knowledge of performer competence, and knowledge of performance outcome on performance error identification," *Research Quarterly*, vol. 50, pp. 318–327, 1979.

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

[3] S. Bronner, and S. Ojofeitimi, "Pelvis and hip three-dimensional kinematics in grand battement movements," *Journal of Dance Medicine and Science*, vol. 15, pp. 23-30, 2011.

[4] J. M. Shippen, and B. May, "Calculation of muscle loading and joint contact forces during the rock step in Irish dance," *Journal of Dance Medicine and Science*, vol. 14, pp. 11-18, 2010.

[5] J. C. Chan, J. C., Leung, H., J. K. Tang, and T. Komura, "A virtual reality dance training system using motion capture technology," IEEE Trans. on Learning Techn. Vol. 4,, no. 2, pp. 187-195, 2011.

[6] E. Ho, J. Chan, T. Komura, and H. Leung, "Interactive partner control in close interactions for real-time applications," *ACM Trans. on Multimedia Comp., Comm., and Appl.*, vol. 9,no. 3, pp. 21, 2013.

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

[8] A. Shimada, and R. I. Taniguchi, R., "Gesture recognition using sparse code of hierarchical SOM,". In *IEEE International Conference on Pattern Recognition*. pp. 1-4, 2008.

[9] G. Pierris, and T. S. Dahl, "Compressed sparse code hierarchical SOM on learning and reproducing gestures in humanoid robots,". In *IEEE RO-MAN*. pp. 330-335, 2010.

[10] A. P. Sangole and A. Leontitis, "Spherical self-organizing feature map: an introductory review. In *International Journal of Bifurcation and Chaos*, vol. 16, no. 11, pp. 3195-3206, 2006.

[11] N. H. Dardas, and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. on Instrumentation and Measurement,* vol. 60, no. 11, pp. 3592-3607, 2011.

[12] E. Kavakli, S. Bakogianni, A. Danuabajus, M. Loumou, and D. Tsatsos, "Traditional dance and e-learning: The WEBDANCE learning environment. In *International Conference on Theory and Applications of Mathematics and Informatics*. pp. 272–281, 2004.

[13] P. T. Chua, R. Crivella, B. Daly, N. Hu, R. Schaaf, T. Ventura, J. Camill, J. Hodgins, and R. Pausch, "Tai Chi: Training for Physical Tasks in Virtual Environments," In Proc. *IEEE Virtual Reality*, pp. 87–94, 2003.

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

[15] U. Yang, and G. J. Kim, "Implementation and evaluation of "just follow me": An immersive, VR-based, motion-training system," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 3, pp. 304–323, 2002.

[16] J. C. Chan, H. Leung, J. K. Tang, and T. Komura, "A virtual reality dance training system using motion capture technology," *IEEE Trans. on Learning Technologies*, vol. 4, no. 2, pp. 187–195, 2001.

[17] D. Alexiadis, P. Daras, P. Kelly, N. E. O'Connor, T. Boubekeur, and M. B. Moussa, "Evaluating a dancer's performance using Kinect–based skeleton tracking," In *ACM Multimedia*, pp. 659–662, 2001.

[18] M. Naemura, and M. Suzuki, "A Method for estimating dance action based on motion analysis," In *Computer Vision and Graphics*. Springer Netherlands, pp. 695–702, 2006.

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

[20] D. A. Becker, and A. Pentland, "Using a virtual environment to teach cancer patients T'ai Chi, relaxation, and self–Imagery," In *Proc. Int. Conference on Automatic Face and Gesture Recognition*, 1996.

[21] K. A. Ericsson, R.T. Krampe, and C. Tesch-Roemer, "The role of deliberate practice in the acquisition of expert performance," *Psychological Review*, vol. 100, no. 3, pp. 363–406, 1993.

[22] A. P. Sangole, and A. Leontitis, "Spherical Self-Organizing Feature Map: an Introductory Review," *Inter. J. of Bifurcation and Chaos*, vol. 16, no. 11, pp. 3195–3206, 2006.

[23] G. Caridakis, K. Karpouzis, A. Drosopoulos, and S. Kollias, "SOMM: Self organizing Markov map for gesture recognition," *Pattern Recognition Letters*, vol. 31, no. 1, pp. 52-59, 2010.

[24] D. Fu, Y. Q. Zou, D. Zou, and G. Xuan, "JPEG steganalysis using empirical transition matrix in block DCT domain," *International Worksshop on Multimedia Signal Processing*, 2006.

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