A Machine Intelligence Approach to Virtual Ballet Training

Paisarn Muneesawang
Naresuan University

Naimul Mefraz Khan
Ryerson University

Matthew Kyan
York University

R. Bruce Elder and Nan Dong
Ryerson University

Guoyu Sun and Haiyan Li
Communication University of China

Ling Zhong
Guangdong University of Technology

Ling Guan
Ryerson University

The proposed framework offers real-time analysis and visualization of ballet movements performed in a virtual reality environment. Students receive quantitative assessments—delivered using concurrent, localized visualizations—and a performance score based on incremental dynamic time warping.

The use of technology in dance training has evolved rapidly over the past decade. For many years, the principal method of teaching dance was the use of demonstration-performance exercises. The instructor would give a demonstration, and the student would attempt to imitate the dance while the instructor closely scrutinized the student's movements and then provided feedback. Almost three decades ago, video became widely used to pass choreography down to new generations of dancers. More recently, video databases have begun playing a more active role in dance education—one example is the *Dance in Video* world dance collection (see http://alexanderstreet.com/node/106). Video analysis software has also been introduced that lets researchers retrieve information about dancers' location in a 3D space. Although none of these technologies have replaced the teacher-demonstrator paradigm, they have been an effective means of information storage and retrieval.

In the last decade, researchers have started exploring other ways to utilize digital media in dance training. For example, Second Life avatars have been used as assistants or supplements to demonstrate basic choreographic sequences.[1] Computer-based systems also offer a more quantitative method of dance training than traditional dance teaching, which relies on an instructor's subjective impressions. A number of research works have described quantitative methods for objectively and systematically analyzing and assessing ballet techniques. These methods use video or 3D motion-tracking systems to capture kinematic and kinetic data, and assessment methods have been developed to evaluate the biomechanical properties of the acquired human movement data. The measurements generated by such systems can be used to evaluate the technical development of individual dancers.

The value of these measurements depends entirely on the representational validity of the characteristics selected for the feature set—that is, how well the set of features selected reflects the dance gestures' most aesthetically relevant dynamic properties. The value also depends on the accuracy of feature extraction. Such measurements facilitate the presentation of nonquantitative feedback, which might take the form of a visual comparison of virtual characters[2] or the synthesis of dance partners "on the fly."[3]

This article extends efforts to enhance the traditional teacher-centered, demonstrate-imitate mode of instruction using a ballet training application. Unlike previous works, we integrate automated training exercises for repetitive tasks with highly accurate analytical tools, including automated gesture recognition and assessment. The goal is to provide adequate feedback to facilitate self-driven learning in an immersive environment.[3] Specifically, we developed a machine intelligence method that provides student dancers with an automated assessment of their performances, relative to a virtual instructor, using a 3D Cave Automatic

1070-986X/15/$31.00 © 2015 IEEE Published by the IEEE Computer Society

Virtual Environment (CAVE) to provide the dancers with enhanced feedback and spatial awareness of their movements relative to the virtual instructor.

System Architecture and Contributions

The architecture of our system includes four components: a Kinect sensor, CAVE, gesture recognition, and visual feedback (Figure 1). To enable dance gesture recognition, we offer a new method for analyzing movement trajectories—the spherical self-organizing map (SSOM). Compared with previous works that implement a self-organizing map (SOM) to quantify movement features,[4,5] we adopt the SSOM because its spherical structure provides a more uniform distribution of features during the quantization process.[6]

Our first contribution to dance analysis is to use a Markov empirical transition matrix to analyze movement phrases (or SSOM transitions between cadence points). We also present a Bayesian method for resolving continuous movement into segments in real time. To our knowledge, no previous studies have incorporated these methods.[7,8]

Our second contribution is the development of a complete virtual reality (VR) dance training system that includes gesture recognition, dance assessment, and visual feedback. To date, research has generally focused on the visualization phase. Much emphasis has been placed on rote learning and repeatedly mimicking the dance teacher, so much so that quantitative measures and feedback are crude or nonexistent. However, repetition without feedback does not necessarily result in improved performance.

Some recent papers have proposed an alternative to this rote-learning paradigm, with student assessment conducted rapidly and fed back to the student using a standard automated protocol.[2,9,10] Students are apprised of the accuracy of their performance, and the specific areas performed inaccurately are identified. The proposed system accommodates all the requirements that arise in connection with standard methods of teaching elementary ballet.

Background and Related Work

Here, we outline the basics of our approach to technologically enhanced teaching of elementary ballet. We also compare our approach to related methods in gesture segmentation and recognition.

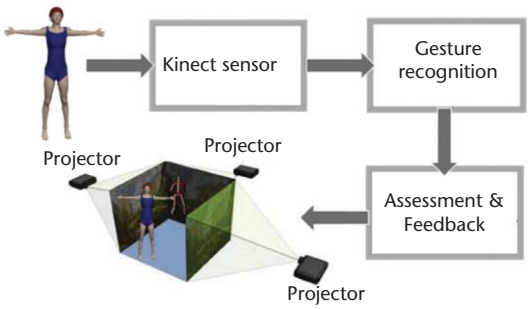

Figure 1. Virtual reality (VR) dance training system architecture. Our system incorporates four components: a Kinect sensor, CAVE, gesture recognition, and visual feedback.

Segmentation, Recognition, and Evaluation

To evaluate partial movements within a dance sequence, gestures must first be detected and isolated (segmented) for comparison. This in turn lets us compare attempted dance movements with the desired performance (from the teacher). In domain-agnostic approaches (those not specific to a particular type of movement under consideration), unsupervised learning is typically used to quantize the possible set of postural states (cadence points) that would be expected within a movement phrase. These states would then be used to train classifiers to detect their occurrence from a temporal stream of gesture data.

Atsushi Shimada, Manabu Kawashima, and Rin-ichiro Taniguchi have used sparse code from SOMs to achieve segmentation and recognition.[4] The latest of these methods performs early recognition of gestures from sparse codes (postures encountered in the SOM for a given movement phrase).[4] In related state-of-the-art methods, k-means clustering has been used to derive bag-of-words (BoW) features that are fed into a multiclass support vector machine (SVM)[7] to achieve gestural classification. Similar BoW decompositions have been combined with stochastic linear formal grammars in dynamic analysis of hand gestures for sign language recognition.[11] Sparse coding has also been used to formulate histogram-based features for multiview human activity recognition from video across distributed camera networks.[12] In these works, stochastic properties of sparse features are often considered at the expense of temporal structure.

Given an isolated movement phrase, the quantitative measurement and evaluation of a dancer's performance level have been proposed

in the literature.[8,9,10] Motion features extracted from video with rhythm elements of dance actions[9] have shown a strong correlation with subjective evaluations of performance. Dynamic time warping (DTW), with exponential time-space scaling,[8] has also been used to compare recognized sequences against known templates. Such methods face difficulties as a result of the noise present in the signal (performance) because there is such high variability when different people attempt to replicate a given motion.

Motor Learning and Feedback

Until recently, the most common means for providing feedback about performance errors involved a high degree of abstraction, presenting errors in the form of simple plots, gauges, bars, lines, or numbers.[13] Abstract visualizations might suffice for simple tasks because they can represent a key feature of a movement in an unambiguous way, but few art/dance students relate to them immediately and emotionally. Furthermore, it is difficult to convey to dance students feedback about complex multidimensional movement in 3D space in abstract form. They are accustomed to learning through an immediate sensory (kinesthetic) experience.

Augmented reality and VR simulators have great potential for facilitating motor learning. Feedback can be provided either during task execution (concurrent feedback) or afterward (terminal feedback). Although terminal feedback is effective for simple tasks, concurrent feedback is more effective for complex tasks with several degrees of freedom, which are unlikely to be mastered in a single session.

Concurrent visual feedback presents side-by-side or superimposed moving-image visualizations of the ideal movement[14] and student's actual performance. The concurrency of the feedback makes the relevant information more immediately comprehensible. Concurrent visual feedback has been shown to be effective in learning complex tasks in domains such as sports and physical rehabilitation. (For example, superimposing target angles over live-action video has been successfully used to teach pitching in baseball.) Similarly, concurrent display bars or force-time plots have been used to indicate the deviation from the desired or ideal (target) force for physical therapy patients practicing mobilization skills. In some cases, a head-mounted display provides concurrent visual feedback by superimposing a ghost target image over what the client would ordinarily see. Studies found that this form of feedback was not optimal however because it required frequent head movements and restricted the user's field of view.

Prescriptive feedback is designed to help more experienced learners correct deviations between their performance and the target movement. Such learners benefit from more focused feedback, with the information supplied when deviations reach a certain threshold. With such feedback, a convention is generally used to signify the degree of movement errors (such as highlighting with different colors superimposed on the limbs). Prescriptive feedback allows learners to focus on the specific aspects of their performance they want to correct. In this respect, prescriptive feedback based on color bar animations has an advantage over real videos (as well as side-by-side and superimposed visualizations) in that the animations are simplified to the point of offering only the most salient information.

Proposed System

Our training system is a unified framework that uses the CAVE VR environment to offer students visual feedback based on computer analyses of their performances. The following is a general overview of the framework:

- Students either watch a virtual teacher demonstrate a dance sequence and attempt to repeat the movement phrase, or they perform the phrase without watching the virtual teacher, and the system automatically recognizes the movement phrase.
- When students have completed a performance, the gesture recognition engine is activated and the gesture data for the teacher's performance of the corresponding phrase is called up and sent to the system's feedback component.
- Students examine the feedback provided in an immersive 3D environment.

The system also provides visual feedback "on the fly," as each student repeats a phrase. When a sequence of gestures (already known to the system) are performed, the system offers concurrent feedback (in side-by-side, superimposed, or descriptive form) to give the student a real-time analysis of the performance.

Unsupervised Gesture Parsing

We use the term "dance posture" to refer to a particular configuration of body parts at a moment in time; these postures can be thought of as cadence points in movement phrases. For the system, these postures are represented by a particular state of sensor values at a moment in time. Gestures involve temporal data, presenting a context for the posture that situates it in relation to the states that have led to or follow the present time step. Such a collection of states and their layout can be indicative of a movement phrase for the gesture recognition process.[14] We employ an unsupervised learning strategy to build an informative posture space, and we characterize the temporal layout of postures in the space for gesture recognition.

The gesture recognition process begins with automatically parsing samples into a discrete set of postures, drawing on a spectrum of expected gestures. We adopt the SSOM demonstrated by Archana P. Sangole and Alexandros Leontitis for this process.[6] Figure 2a shows the SSOM structure, where the cadences or postures are projected (quantized) onto the map. The six basic positions of ballet (*bas bras* and the positions 1 through 5) are also shown in Figure 2a. Given the six postures $P_1 - P_6$, a distinct gesture G_{ij} is formed as the dancer moves from posture P_i to posture P_j.

When using a SSOM for parsing, the discrete space is constructed so as to retain associations that exist in the original input space—that is, postures are positioned on the map near other similar postures. As a consequence of this topology-preserving mapping, a sequence of expected "posture moments" or cadences within an ongoing flow of movements could be expected to trace a comparatively smooth trajectory on the map. From this trajectory, we formulate descriptors for each gesture.

The goal of unsupervised learning is essentially to discover significant patterns in a given set of data. The patterns are usually stored as a set of clusters or groups of similar data. We form clusters using the SSOM algorithm. The SSOM lattice forms a closed-loop sphere (see Figure 2a). Thus, the neighborhood learning allows learned postures to be allocated to, and distributed across, nodes on the lattice. Each training pattern in the input space is connected to every clustered unit by weight vector $\mathbf{w}_{i,j,k}$, a key posture from the input space, where (i, j, k) are the indices in 3D space. The total number of nodes represents the number of postures the map can learn. In this representation, nodes are each equidistant from their immediate neighbors, with which they form a hexagonal neighborhood.

The SSOM learning phase is the procedure for formulating weight vectors $\mathbf{w}_{i,j,k}$, using an adaptive rule:

$$\Delta \mathbf{w}_{(i,j,k)^*} = \alpha[\mathbf{x}_t - \mathbf{w}_{(i,j,k)^*}],$$

where $\mathbf{w}_{(i,j,k)^*}$ is the weight vector of the winning node $(i, j, k)^*$, α is the learning rate, and $\mathbf{x}_t$ is the current input at time t. Nodes compete to see which is most representative of a given input pattern presented to the network. A node is identified as the winner if its weight vector minimizes Euclidean distance over the set of all nodes.

Trajectory Analysis

In addition to the six postures mapped in Figure 2a, we map four instances of gesture G_{61} and show the outcome in Figure 2b. This visualization of the SSOM and the gesture trajectories on it show that even differences in the duration of the gestures do not appear to impact the consistency with which the gestures map onto the posture space. All gestures appear to trace characteristic, repeatable paths on the unit sphere. The consistency of the mapping indicates there is a high degree of stability in the representation of the gestures, and this in turn suggests there is sufficient overlap when extracting a robust descriptor for each gesture category. In the literature, where mapping is based on hierarchical SOM[5] or hierarchical k-means,[7] the sparse code and BoW descriptors are usually explored. Specifically, the indices of winning nodes are obtained, and the following descriptors are constructed.

- *Sparse code* results in identifying a pattern of activated winner nodes for a gesture element. A sparse code vector only indicates the existence of a set of postures, not their frequency of occurrence.
- *Posture occurrence* is analogous to the popular BoW model. It is obtained by aggregating the occurrence of postures in a gesture against the indexed set of nodes on the map.

These descriptors do not consider the temporal arrangement of postures or cadences in

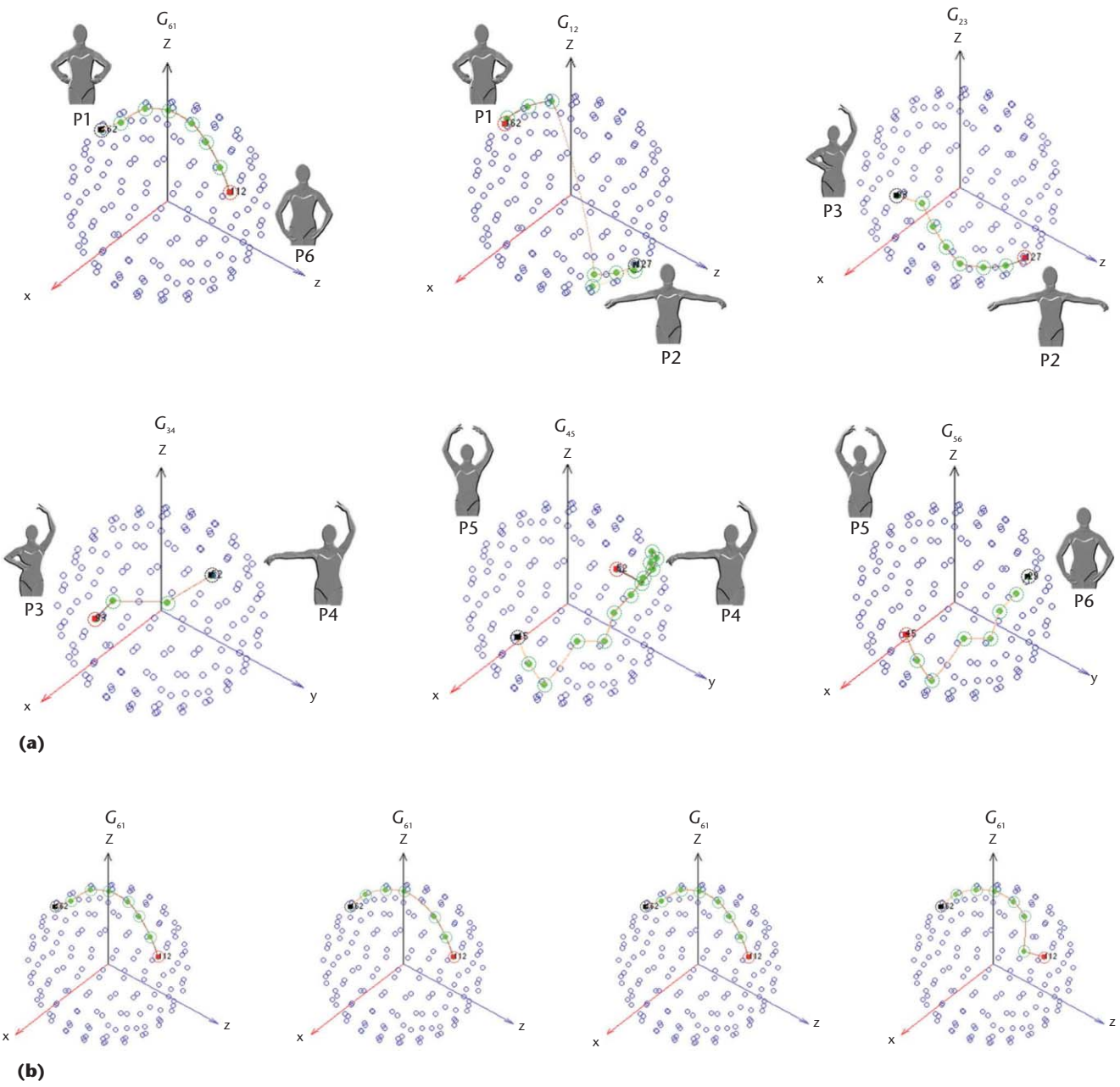

Figure 2. Gesture projections for ballet dance. (a) The six postures are mapped onto the posture space. Each gesture represents as a trajectory on the spherical self-organizing map (SSOM). (b) Four instances of gesture G_{61}. Smooth, local sets of postures show stable, highly repeatable trajectories.

the map. They consider only the occurrence of map units and the frequency of the individual nodes for constructing descriptors. We observe that these methods strive to maintain the marginal histogram of the SSOM indices (first-order statistics). Because a gesture contains cadences (postures) that strongly correlate with their neighbors, the adoption of second-order statistics, such as covariance and co-occurrence matrices, are more appropriate for capturing the dependency between pairs of cadences from the SSOM trajectory. This lets us obtain a descriptor using *posture transition* (PT).

We define a trajectory as a set of indices of (winning) map units and model it using a Markov random process. To capture the dependencies between SSOM nodes in the trajectory, the horizontal Markov empirical transition matrix of the dataset is calculated. The matrix's element is given by the probability

Input: gesture sequence $S_u = [x^{(t_0)}, \ldots, x^t, \ldots, x^T]$
Output: $P_t(c|h_s)$; $argmax_{c,t}\{P_t(c|h_s)\}$

Set $t = t_0 = 0$;
Repeat
Let input gesture $S_u = \left[x^{t_0}, \ldots, x^t\right]$
Let h_s be the PT template calculated from S_u, and let h_c be the PT template of the cth gesture class
Calculate the likelihood, $P_t(h_s \mid c) = HI(h_s, h_c) = \sum_i \min[h_{s,i}, h_{c,i}]$

Calculate the prior, $P_t(c) = \begin{cases} \frac{1}{C}, & \text{if } t = t_0 \\ \frac{P_{t-1}(c|h_s) \cdot HI(h_s, h_c)}{\sum_c P_{t-1}(c|h_s) \cdot HI(h_s, h_c)}, & \text{otherwise} \end{cases}$

Calculate the posterior probability $P_t(c \mid h_s)$ for all c,

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

If $max[P_t(c \mid h_s)] > T_1$ (let T_1 = threshold)
$t_0 = t$
Reset prior $P_t(c) = 1/C$
Recalculate posterior $P_t(c \mid h_s)$

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

Figure 3. Algorithm 1 for online gesture recognition. Let S_u be the input posture sequence (captured by the sensor), and let c= 1, …, C be the indices of gesture classes known by the system. Given S_u, the system estimates the posterior probability of the gesture class c, which we denote as $P(c \mid S_u)$, using the following rule: Posterior $\propto$ Prior $\times$ Likelihood.

$$P_h(u_{i+1} = n \mid u_i = k) = \frac{\sum_{i=1}^{W-1} \delta(u_i = k, u_{i+1} = n)}{\sum_{i=1}^{W-1} \delta(u_i = k)},$$

where δ is the delta function; u_i and u_{i+1} are the respective indices of a pair of neighboring nodes; W is the size of the trajectory, $k,n \in \{1, \ldots, M\}$; and M is the number of nodes in the SSOM. The PT feature vector is formed by arranging the elements of this matrix P_h, $h = 1, \ldots, M^2$ into a 1D template.

Isolated Gesture Recognition

We use the term *balletic sequence,* or simply sequence, to refer to a linked sequence of balletic gestures. Our system can segment sequences into isolated gestures. We use a probabilistic framework for recognition and adopt a simple Bayesian formula for progressively estimating an updated posterior probability. The pseudocode for online gesture recognition is described by Algorithm 1 (see Figure 3). Let S_u be the input posture sequence (capture by the sensor), and let $c = 1, \ldots, C$ be the indices of gesture classes known by the system. Given S_u, the system estimates the posterior probability of the gesture class c, which we denote as $P(c \mid S_u)$, using the following rule: Posterior $\propto$ Prior $\times$ Likelihood.

Concurrent Feedback Design

For concurrent feedback, there are two training phases.

First-phase training. Once the system has identified the best gesture class matches for a given performance, the remaining problem is to determine how well the student has performed the dance phrase compared with the teacher's version. We provide that comparison visually, using concurrent feedback. Specifically, we provide

(a) (b)

Figure 4. VR dance training with feedback protocols. (a) Side-by-side feedback and (b) superimposed feedback.

both side-by-side (Figure 4a) and superimposed (Figure 4b) comparisons of the student and teacher performances. While the student performs the dance phrase, the system provides options for the student and teacher models to face one another, have their backs to the audience, or face the audience. (The latter is generally the most useful.)

It is worth noting that CAVE displays differ significantly from 2D videos. In the CAVE system, students wear stereo glasses with optical markers, which allows for 3D visualization. Moreover, the location and orientation of the user's viewpoint (head pose) is tracked, and this data is used to determine what is displayed on the screens. This tracking offers greater freedom of interaction and, accordingly, more effective feedback.

Second-phase training. In the first phase of dance training, students use a watch and imitate strategy. In this second phase, side-by-side and overlay (or superimposed) feedback provide the student with visual and kinetic cues. We take special care in this phase to give to the learner only the most salient information to prevent cognitive overload. This consideration led us to use prescriptive feedback. To visually convey performance measures concurrently, we use a skeleton silhouette. Specifically, we use an incremental DTW (IDTW) algorithm[15] to compare the two dance performances. The IDTW score for each joint is calculated separately and mapped to a color table, which is used to render the colored skeleton silhouette. The color of a limb represents the IDTW distance (that limb's degree of conformance with the template).

We use the skeletal feature data from the teacher example as a benchmark and compare it in real time with the student's performance. Incorporating IDTW into the algorithm to calculate a distance score between performances makes it possible to compare an incomplete sequence with a benchmark of a complete sequence. Because the IDTW can evaluate a partial performance (the student's) against the complete sequence (the teacher's), it allows for an incremental similarity score, substantially improving computational efficiency over the traditional DTW algorithm. Pseudocode for the IDTW algorithm is presented in Algorithm 2 (see Figure 5).

The final aspect of the system concerns communicating performance measurements to the student. Here, we implement a per limb procedure (we loosely refer to a skeletal segment connecting neighboring joints as a limb). Rather than calculating the IDTW score as a whole, we calculate separate IDTW scores for each joint and then convert these into scores for each limb. We map the IDTW score to a color table and then convey our feedback on the performance using the virtual skeleton.

To prepare the visual evaluation, we represent IDTW scores for each joint by $D^l_{IDTW}, l = 1, \ldots, K$, where K is the total number of joints. We calculate the performance at a particular limb by

$$Z = \exp\left(-\upsilon \frac{(D^{l_a}_{IDTW} + D^{l_b}_{IDTW})}{2}\right),$$

where l_a and l_b denote the joints that form this particular limb, and v is a parameter to control the sensitivity of the performance measure. The

```
Input: U - The user sequence up to current time (length N)
       E - The expert (full) sequence (length M)
       G - M × N cumulative cost matrix up to current time
       V - Next frame in user sequence
Output: Updated IDTW distance

  1.  Function IDTW(U, E, G, V)
      Q ← (N + 1)
      U_Q ← V
      G(1 … M, Q) ← array(1 … M)
      for i ← (max(1, Q), min(M, Q)) do
              G(i,Q) ← min(G(i−1,Q), (G(i − 1, Q − 1), (G(i − 1, Q − 2)) +
      d(U_Q, E_i)
      end for
      return min(G(1 … M,Q))/Q
  2.  End function
```

Figure 5. Algorithm 2. The Incremental DTW algorithm, where **$d(U_Q, E_i)$ denotes the Euclidean distance between joints in frames U_Q and E_i.**

value of Z is projected to a color map in blue, aqua, green, yellow, or red. The color stays closer to blue if the student is doing well and shifts toward red as the performance worsens.

Experiments

Three subjects (one teacher and two students) participated in our detailed experiments. Nine other subjects participated in recording of continuous dance sequences for additional quantitative evaluation.

System Configuration

The CAVE system we use consists of four stereoscopic channels, each with a projector and screen. All channels are driven by a graphics cluster of five nodes; one node serves as the cluster master, while the other four each drive a single channel. The student wears active stereo glasses with optical markers that are tracked by a six degrees-of-freedom tracking system. When the student moves, the glasses' position and orientation are tracked by an array of cameras distributed above the screens. The system uses the tracking data to determine the content to be displayed on each of the screens. The Unity 3D game engine was used to implement visual feedback, translated by Middle VR for display in the CAVE.

The real-time analysis system extracted features from student dancers. Each frame recorded by the Kinect's sensor contains information about the location (in 3D) of 20 joints in the figure it sees. The joint locations are calculated relative to the hips (that is, normalized to form a unit vector emanating from the hip). Information concerning joint locations must be scaled for the process to accommodate users with different limb-to-limb ratios. We used these normalized joint locations as information to be fed into the subsequent steps in our procedures to identify a particular gesture and generate feedback (using the IDTW algorithm).

We considered three spherical configurations for the SSOM used for training and representing posture space. Each of these maps—C1, C2, and C3—has a different icosahedron decomposition level (relating to the number of map nodes distributed over the sphere). These maps result in 42, 162, and 642 nodes (for C1, C2, and C3, respectively).

Recognition and Evaluation

We started with the six basic positions of traditional ballet (*bas bras* and positons 1 through 5) and constructed 30 gestures, which include the reversal movements.[14] For the SSOM training process, the system recorded the teacher performing each gesture 10 times, resulting in 300 instances of dance gestures. When training the SSOM, five of 10 instances of each gesture class were randomly selected to form gesture templates, and then all 300 were classified against these benchmarks. Table 1 shows the recognition result for teacher data.

The results show that PT and PTSC (PT and its sparse code) are more robust than other methods discussed, regardless of the SSOM configuration. We also observe that the higher number of SSOM nodes, the higher the

Table 1. Recognition accuracy results. The system was tested with different configurations for the spherical self-organizing maps (SSOMs).

Framework	Clustering method	Indexing method	Classifier	Average recognition accuracy (%)		
				Teacher (300 instances, 30 classes)	Student 1 (300 instances, 30 classes)	Student 2 (300 instances, 30 classes)
Proposed	SSOM (42 nodes)	Posture occurrence (PO)[7]	Template matching	85.0	75.1	69.7
		Posture sparse code (PSC)[5]	Template matching	57.3	49.9	51.7
		Posture transition (PT)	Template matching	95.3	91.6	83.4
		Posture transition sparse code (PTSC)	Template matching	92.3	82.6	82.1
	SSOM (162 nodes)	PO	Template matching	89.0	75.1	73.3
		PSC	Template matching	67.3	65.7	53.5
		PT	Template matching	98.3	93.4	88.8
		PTSC	Template matching	94.7	93.7	83.0
	SSOM (642 nodes)	PO	Template matching	90.0	80.3	79.5
		PSC	Template matching	83.7	77.6	75.1
		PT	Template matching	99.0	93.7	**92.4**
		PTSC	Template matching	**99.7**	**95.0**	92.1
Sparse code of SOMs[4]	SOM (625 nodes)	PSC	Template matching (Hausdorff distance)	93.33	80.1	74.63
Bag of words and support vector machine[7,11]	K-means (625 nodes)	PO	Multiclass SVM	99.33	29.33	13.0
Sparse code[12]	K-means (625 nodes)	PSC	Multiclass SVM	100.0	11.0	6.67

performance of the recognition module. The average recognition rate is close to 100 percent, justifying our observation that incorporating temporal information from gestures into our recognition module improves the performance of the PT and PTSC.

Our system was trained to identify the dance gestures performed by the students using data from performances by the teacher. Like the teacher, the students performed each gesture 10 times, and each performance was recorded, resulting in 600 samples for testing. Table 1 shows that the system's recognition of previously unseen student gestures is high when our PT and PTSC indexing protocols are implemented. We also observe that the system has different recognition rates for the two students, in part because of how much they deviated from the template.

We replaced the significant gesture recognition using different methods to report their effects on the system. The first method we considered for comparison utilized the sparse code of the SOMs.[4] In the training phase, a SOM was used to learn the postures that were the elements of all (teacher) gestures. Then, the sparse code was used to represent a gesture pattern by a set of a small number of nodes. In the testing phase, we used the Hausdorff distance to measure the similarity between the sparse code of an unknown input pattern and the sparse code of training patterns.

The second method we considered for comparison was based on using the BoW feature, k-means clustering, and a SVM. As in other work,[7,10] we employed a method for recognition of dance gestures via BoW features and multiclass SVM. In the training stage, after extracting

Table 2. Average similarity, F_1, recall, and precision values of the proposed gesture recognition system, obtained from continuous dance sequences (with the following order of gestures: G_{61}, G_{12}, G_{23}, G_{34}, G_{45}, and G_{56}) for nine students.

Students	Ground truth: gesture (averaged number of frames)	Recognition performance (averaged over all gesture classes and students)			
		Similarity	F_1	Recall	Precision
1–9	G_{61} (74), G_{12} (52), G_{23} (52), G_{34} (53), G_{45} (50), G_{56} (53)	0.9885	0.9938	0.9938	0.9947

skeleton points for every training frame (obtained from the Kinect), we used a vector quantization technique, k-means clustering, to map skeleton points from every training frame onto a unified dimensional histogram vector (BoW). This histogram was treated as an input vector for a multiclass SVM to build the training classifier. In the testing stage, for every frame captured by the Kinect, the skeleton points were fed into the cluster model to map them into a BoW vector, which was finally fed into the multiclass SVM classifier to recognize the dance gesture.

Lastly, in the third method,[12] the training and testing processes were the same as that of the second method, except that we used the sparse code instead of BoW.

The last three rows of Table 1 show the results obtained by the three methods. All three methods recognized, with more than 93 percent accuracy, all 300 gesture instances of the teacher's performance. This is easy to justify, as we used only teacher data for training. For the recognition of dances performed by the two students, the performance of the first method[4] (using SOM and PSC) was similar to SSOM and PSC. However, this performance was much lower than that of the proposed method using SSOM and PT. This result shows the superiority of the proposed method for the recognition of all dance gestures that included reversal gestures. The multiclass SVM implemented in the second and third methods[7,11,12] performed with 100 percent accuracy when recognizing the teacher's dances. The multiclass SVM-based recognition methods on the other hand performed poorly when attempting to recognize student dances. In other situations,[7,11,12] where sufficient training sets were available, the SVM method may perform effectively. In the current application, we used only the teacher's performances to train the SVM classifiers; for every gesture class, we used 10 instances of positive samples and 290 instances of negative samples to train the SVM. The inadequacy of this training set may be unfavorable for applying the SVM method to dance training system.

We thus see that the proposed recognition method is the most suitable approach for the ballet training system because it attained an accuracy of 95 and 92 percent when recognizing the student 1 and 2 dances, respectively.

Isolation of a Gesture from a Continuous Sequence

To assess our system's ability to recognize and isolate gestures from a sequence, we asked nine students to perform (continuously) a linked series of dance gestures. We then applied the criterion of maximum posterior probability (Algorithm 2, Figure 5) to extract each successive gesture in order to estimate its duration. We assessed the performance of the proposed online gesture recognition method through quantitative evaluation.

For this evaluation, we applied recall, precision, F_1, and similarity metrics[16] to continuous gesture sequences of the nine students. F_1 and similarity are derived from recall and precision and are considered a significant measurement of accuracy. Table 2 shows the average accuracy rates of the proposed online gesture recognition method for all metrics applied to all the students' performance. With some minor noise at the beginning and end of the gestures (and dances), the selection of gesture class appears to follow the actual sequence (F_1 producing 99.4 percent, and similarity 98.9 percent, averaged over all test data). Clearly, the findings reflected in Table 2 indicate that the proposed method achieves superior gesture recognition results for all gesture classes in all gesture sequences assayed.

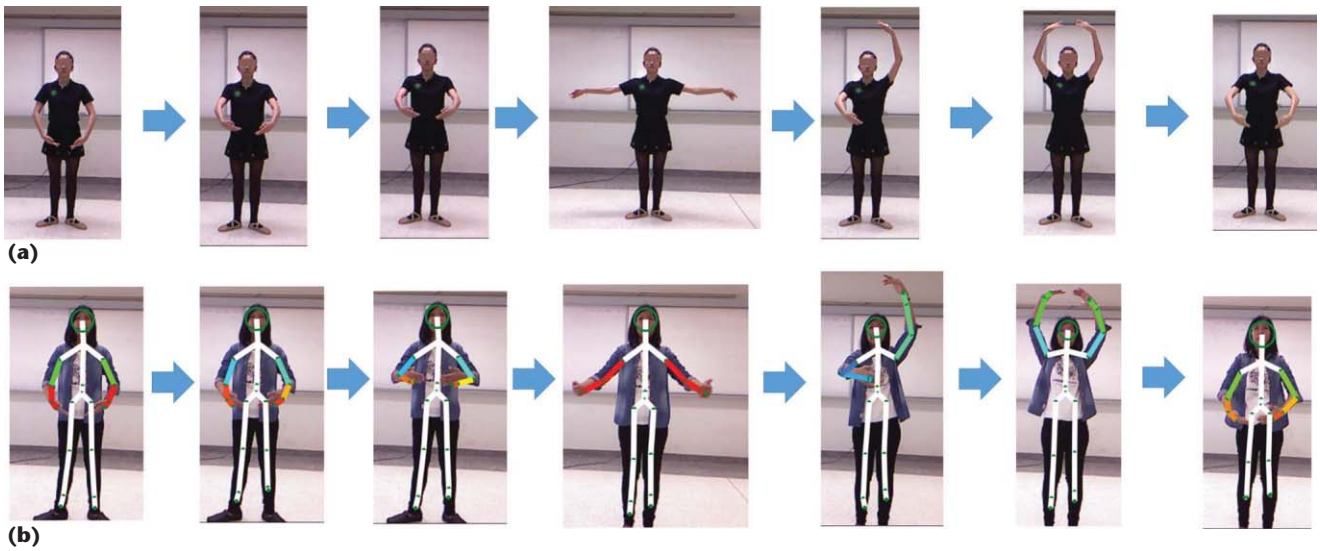

Figure 6. Descriptive feedback presented to a student while performing a dance sequence, based on a comparison with (a) the teacher's performance. (b) The student receives feedback in the form of colors of the skeleton.

Visual Feedback Result

The implementations of concurrent feedback using side-by-side and superimposed images are illustrated in Figure 4. A detailed discussion of these results is available elsewhere.[14] Here, we present only the experimental result, including comments on the descriptive feedback the system offers, based on the IDTW algorithm.

The bottom row of images in Figure 6 shows the feedback offered in real time during a practice/teaching session. The student executes a dance sequence and receives visual feedback in real time, allowing her to examine her performance, using information provided by the color of the skeleton. In the figure, the performance of the student is compared with the teacher's, over selected frames. For assessments that use the IDTW algorithm, the system applies a joint filtering process to identify active elements, which are presented in colors other than white, while presenting inactive joints in white.

As we can see, the proposed system can report the mistakes made by the student in an easy-to-interpret manner. When the student performs well, the color of the visual feedback stays close to blue. But when the student's performance deviates from the teacher's, the system is able to highlight where he/she is going wrong. For example, in the last posture/cadence (see the last frame in Figure 6), the performer is not positioning her arms correctly. So the color has shifted toward green/yellow. Also, in the middle frame in Figure 6, the red color marks the incorrect placement of both hands. This visual interpretation helps students by immediately alerting them to matters that need attention.

We measured the total IDTW score for each gesture to compare the two student dancers (the students performed the six dance gestures reported in Table 1). The first student performed reasonably well, while the second performed did not do as well. After all the performances were completed, the first student had an average IDTW score of 0.202 ± 0.0887, whereas the second student's average IDTW score was 0.586 ± 0.2417. The high IDTW score and standard deviation signify a weaker performance by the second student.

We hope in future work to overcome some of the limitations of our present study. In our experiments, we tested the system for the recognition of the six rudimentary positions of basic ballet and transitions between them. We intend to move beyond the traditional content of the first two or three dance classes, something that should be possible because of elementary ballet's restricted vocabulary of movements. The syntax governing these movements and their connection is developed partly from natural limitations on combining movement and partly on convention. When we enlarge the set of postures beyond that rudimentary set taught in the first classes in elementary ballet, it becomes important to capture the whole skeleton correctly. The first two versions

of the Kinect require dancers to face the camera; furthermore, they do not capture correctly postures that involve bending backward. We also observed that the Kinect sometimes detected the leg joints inaccurately. These problems might be solved by employing cameras that use time-of-flight measurements, such as the Kinect 2 or other sophisticated data-capturing devices. **MM**

Acknowledgments

This special issue is a collaboration between the 2014 IEEE International Symposium on Multimedia (ISM 2014) and *IEEE MultiMedia*. This article is an extended version of "An Advanced Computational Intelligence System for Training of Ballet Dance in a CAVE Virtual Reality Environment," presented at ISM 2014.

References

1. C. Hong et al., "Transcultural Perspective on Digital Practices and the Arts in Higher Education," *Dance Dialogues: Conversations Across Cultures, Art forms and Practices*, World Dance Alliance Global Summit, 2008; http://core.ac.uk/display/10895770.
2. J.C. Chan et al., "A Virtual Reality Dance Training System Using Motion Capture Technology," *IEEE Trans. Learning Technology*, vol. 4, no. 2, 2011, pp. 187–195.
3. E. Ho et al., "Interactive Partner Control in Close Interactions for Real-Time Applications," *ACM Trans. Multimedia Computing, Communications, and Applications*, vol. 9, no. 3, 2013, article no. 21.
4. A. Shimada, M. Kawashima, and R. Taniguchi. "Improvement of Early Recognition of Gesture Patterns Based on a Self-Organizing Map," *Artificial Life and Robotics*, vol. 16, no. 2, 2011, pp. 198–201.
5. G. Pierris and T.S. Dahl, "Compressed Sparse Code Hierarchical SOM on Learning and Reproducing Gestures in Humanoid Robots," *Proc. IEEE Int'l Conf. Robot and Human Interactive Comm. (RO-MAN)*, 2010, pp. 330–335.
6. A.P. Sangole and A. Leontitis, "Spherical Self-Organizing Feature Map: An Introductory Rev.," *Int'l J. Bifurcation and Chaos*, vol. 16, no. 11, 2006, pp. 3195–3206.
7. 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. Instrumentation and Measurement*, vol. 60, no. 11, 2011, pp. 3592–3607.
8. M. Raptis, D. Kirovski, and H. Hoppe, "Real-Time Classification of Dance Gestures from Skeleton Animation," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, 2011, pp. 147–156.
9. M. Naemura and M. Suzuki, "A Method for Estimating Dance Action Based on Motion Analysis," *Computer Vision and Graphics*, Springer Netherlands, 2006, pp. 695–702.
10. D. Alexiadis et al., "Evaluating a Dancer's Performance Using Kinect-Based Skeleton Tracking," *Proc. ACM Int'l Conf. Multimedia*, 2011, pp. 659–662.
11. M.R. Abid, E.M. Petriu, and E. Amjadian, "Dynamic Sign Language Recognition for Smart Home Interactive Application Using Stochastic Linear Formal Grammar," *IEEE Trans. Instrumentation and Measurement*, vol. 64, no. 3, 2015, pp. 596–605.
12. J. Luo, W. Wang, and H. Qi, "Feature Extraction and Representation for Distributed Multi-View Human Action Recognition," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, 2013, pp. 145–154.
13. M. Eriksson, K.A. Halvorsen, and L. Gullstrand, "Immediate Effect of Visual and Auditory Feedback to Control the Running Mechanics of Well-Trained Athletes," *J. Sports Sciences*, vol. 29, no. 3, 2011, pp. 253–262.
14. G. Sun et al., "An Advanced Computational Intelligence System for Training of Ballet Dance in a CAVE Virtual Reality Environment," *Proc. IEEE Int'l Symp. Multimedia* (ISM), 2014, pp. 159–166.
15. N.M. Khan et al., "A Visual Evaluation Framework for In-Home Physical Rehabilitation," *Proc. IEEE Int'l Symp. Multimedia* (ISM), 2014, pp. 237–240.
16. L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," *IEEE Trans. Image Processing*, vol. 17, no. 7, 2008, pp. 1168–1177.

Paisarn Muneesawang is an associate professor of computer engineering at Naresuan University. His research interest includes machine learning and its applications to multimedia. Muneesawang received his PhD in engineering from the University of Sydney. He has authored and edited five books in multimedia signal processing and machine learning. Contact him at paisarnmu@nu.ac.th.

Naimul Mefraz Khan is a research engineer at Sunnybrook Health Sciences Center. His research interests include machine learning, computer graphics, and computer vision. Khan received his PhD in engineering from Ryerson University. Contact him at n77khan@ee.ryerson.ca.

Matthew Kyan is an associate professor in the Department of Electrical Engineering and Computer Science at York University, Canada. His research

interests include audio-visual signal processing, knowledge-assisted visualization, and immersive computing. Kyan received his PhD in electrical engineering from the University of Sydney, Australia. Contact him at mkyan@cse.yorku.ca.

R. Bruce Elder is a filmmaker; critic; and professor of film, media, and communication studies at Ryerson University. His research concerns the influence of developments in science and technology on vanguard arts; he approaches this topic in a number of ways: as a cultural historian, as a media artist, and as a technology developer. Contact him at belder@ryerson.ca.

Nan Dong is the lab manager and research engineer in the Center for Interactive Multimedia Information Mining at Ryerson University. His research interests include augmented reality, 3D, and multimedia in education. Dong received his MASc in electrical engineering from Ryerson University. Contact him at gdongnan@gmail.com.

Guoyu Sun is a lecturer in the School of Animation and Digital at Communication University of China. Contact her at gysunlu@gmail.com.

Haiyan Li is an associate professor in the Department of Interactive Arts at Communication University of China, Beijing. Her research interest includes human-computer interaction and animation technology. Li received her PhD in communication and information systems from Communication University of China. Contact her at hyli@cuc.edu.cn.

Ling Zhong is a lecturer with the Faculty of Automation at Guangdong University of Technology. His research interests include nonlinear image and video processing, machine learning, and computer vision. Zhong received his PhD in control theory and control engineering from the Guangdong University of Technology. Contact him at tonnyzhong@163.com.

Ling Guan is a professor and a Tier I Canada Research Chair at Ryerson University, Toronto, Canada. His research interests include multimedia processing, machine learning, and adaptive signal processing. Guan received his PhD in electrical engineering from University of British Columbia, Canada. He is an IEEE Circuits and System Society Distinguished Lecturer and the recipient of the 2014 IEEE Canada C.C. Gotlieb Computer Medal. Contact him at lguan@ryerson.ca.