Real-Time Human Movement Retrieval and Assessment With Kinect Sensor

Min-Chun Hu, *Member, IEEE,* Chi-Wen Chen, Wen-Huang Cheng, *Member, IEEE,* Che-Han Chang, Jui-Hsin Lai, and Ja-Ling Wu, *Fellow, IEEE*

***Abstract*—The difficulty of vision-based posture estimation is greatly decreased with the aid of commercial depth camera, such as Microsoft Kinect. However, there is still much to do to bridge the results of human posture estimation and the understanding of human movements. Human movement assessment is an important technique for exercise learning in the field of healthcare. In this paper, we propose an action tutor system which enables the user to interactively retrieve a learning exemplar of the target action movement and to immediately acquire motion instructions while learning it in front of the Kinect. The proposed system is composed of two stages. In the retrieval stage, nonlinear time warping algorithms are designed to retrieve video segments similar to the query movement roughly performed by the user. In the learning stage, the user learns according to the selected video exemplar, and the motion assessment including both static and dynamic differences is presented to the user in a more effective and organized way, helping him/her to perform the action movement correctly. The experiments are conducted on the videos of ten action types, and the results show that the proposed human action descriptor is representative for action video retrieval and the tutor system can effectively help the user while learning action movements.**

***Index Terms*—Feature extraction, human action, human skeleton, motion assessment, nonlinear time warping, video retrieval.**

I. Introduction

FOR the very beginner who wants to learn dance moves or the patient who needs to do rehabilitation exercises everyday, it would be great to have a professional instructor to teach him/her how to perform each action movement correctly. However, it is uneconomical to hire a human tutor every time he/she practices. In recent years, applications developed based on Kinect are getting popular since human poses can be more easily estimated based on the sensed RGB-D information. Dancing games such as LET'S DANCE with Mel B [1], Dance Central [2], and Just Dance [3] are designed for entertaining purposes so that players can have fun from doing basic dance moves or performing freestyle movements according to the designated dance moves. Inspired by theses games, we aim to develop a real-time action tutor system for analyzing the user's movements captured by Kinect and giving detailed motion instructions. Compared to learning with a human tutor/instructor, learning in front of the proposed action tutor system would make the user feel less embarrassed especially when he/she performs wrong movements.

Different from conventional real-time feedback systems based on the Kinect sensor (such as the previously mentioned dancing games), the proposed action tutor system designed for the beginner/patient gives more detailed motion instruction of each body part, and both spatial and temporal correlation among postures are considered in the movement assessment. To be precise, conventional dancing games define several key postures for each action and the correctness score increases as long as the user performs postures spatially similar to the predefined ones in limited time, while the temporal correlation among postures is seldom considered. Besides, as an action tutor, our system automatically reminds the user to learn an action again once it detects an awfully inaccurate movement. In addition to helping the user correctly learn an action movement, our system also facilitates him/her to find the target learning exemplar in a large video database. The user can just perform a query action resembling the target movements in his/her mind, and the system will recommend a list of similar action videos (also captured by Kinect) in the database by the technique of action video retrieval. The user then selects one video from the list as the learning exemplar and follows it to practice. In the mean time, the system gives detailed movement assessment so that the user can perform exercise movements by himself/herself as if accompanied by a private human tutor/instructor.

The most significant challenge of human action analysis is to account for the variations which could highly affect the observations. Sheikh *et al.* [4] explicitly pointed out the possible variability in terms of three transformations.

1) *Viewpoint Transformations:* The position and the orientation of the camera should not affect the analysis results.
2) *Anthropometric Transformations:* An action can be performed by people of different height, weight, or gender. Therefore, the action analysis method should be invariant to human anthropometric ratios.

Manuscript received July 10, 2013; revised January 10, 2014; accepted June 17, 2014. Date of publication July 22, 2014; date of current version March 13, 2015. This paper was recommended by Associate Editor W. Hu.

M.-C. Hu is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: anita_hu@mail.ncku.edu.tw).

C.-W. Chen is with the Application Innovation Consultant with IBM, Taipei, Taiwan (e-mail: euro@cmlab.csie.ntu.edu.tw).

W.-H. Cheng is with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan (e-mail: whcheng@citi.sinica.edu.tw).

C.-H. Chang, J.-H. Lai, and J.-L. Wu are with the Communications and Multimedia Laboratory, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: frank@cmlab.csie.ntu.edu.tw; larrylai@cmlab.csie.ntu.edu.tw; wjl@cmlab.csie.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2335540

3) *Temporal Transformations:* The temporal transformations may be caused by different camera frame rates or different users executing the same action in various speeds. For action recognition/retrieval, we hope the analysis method can tolerate these temporal variations to some extent. In contrast, for movement assessment, we expect to highlight the difference in action execution speed so that the user can perform his/her action with proper speed to prevent sports injuries.

We summarize the technical contributions of this paper as follows.

1) We design a discriminative pose descriptor, including information of the body-joint configuration and the shape/depth distribution of the human silhouette, to reflect the difference between various postures.
2) A novel way to retrieve and learn target exercise movements is proposed with the aid of Kinect, and the retrieval method can deal with the temporal transformation by nonlinear time warping approaches which map the time alignment problem between two multivariant action sequences to the substring finding problem. In most of the existing systems [5], a presegmented step is required to indicate the exact start/end time positions of the actions before calculating pairwise action similarity. By contrast, our system removes this constraint through applying the substring finding technique.
3) We also study the nature and intrinsic limitations of different video matching methods, including cross-correlation alignment, dynamic time warping (DTW) and approximate string matching (ASM). Based on our investigation, a modified version of DTW is proposed to realize action video retrieval.

The remainder of this paper is organized as follows. We briefly introduce related literatures on Kinect and human action analysis in Section II. Section III expounds the framework of the proposed action tutor system from the user's and the system's perspectives, respectively. Section IV depicts the input skeleton model tracked and estimated by the full body analysis middleware of the OpenNI framework [6], [7]. We detail the techniques of similarity measurement and movement evaluation in Section V. The experimental results are shown in Section VI, and conclusions are given in Section VII.

II. Related Work

A. Survey of Kinect

Microsoft released Kinect (a motion sensing device featured with RGB camera, depth sensor, and multiarray) in 2010 and led multimedia applications to a new trend. Kinect is widely used in the industry/research fields of games, robotics, computer graphics, image processing, computer vision, human-computer interaction, and augmented reality [8]. For example, Packer *et al.* [9] utilized Kinect to recognize complex, fine-grained human actions involving the manipulation of objects. Xia and Aggarwal [10] presented a filtering method to extract local spatio-temporal interest points from depth videos captured by Kinect and further build a novel depth cuboid similarity feature to describe the local 3-D depth cuboid. This feature was then applied on activity recognition application [10]. Hsu *et al.* [11] combined Kinect and Bluetooth techniques to build an smart conference system. The Kinect was used as a gesture recognition device to detect each person's skeletons with multifunctions, and a personalized Bluetooth supported equipment was employed to identify each participant's identity. Ren *et al.* [12] applied kernel descriptors, superpixel MRF, and a segmentation tree to achieve scene labeling with RGB-D images. Exploiting the depth information, Liu *et al.* [13] facilitated both camera motion estimation and frame warping to make the video stabilization a much well posed problem. Yang *et al.* [14] took advantage of both color and depth information to predict head pose and generate extra constraints at the face boundary. The face shapes are then tracked based on a nonlinear manifold. Shen *et al.* [15] proposed an exemplar-based method to correct the initial pose estimation result by learning an inhomogeneous systematic bias within specific human action domain. However, Kinect does not always capture reliable depth information and researchers start to improve the depth map captured by Kinect. For example, Shen and Cheung [16] proposed a depth correction and completion algorithm by using depth layers to account for the differences between foreground objects and background scene, the missing depth value phenomenon, and the correlation between color and depth channels. Given a single RGB-D image, Barron and Malik [17] also designed an approach to produce an improved depth map and then applied it to recover intrinsic scene properties from a single image.

B. Survey of Human Action Analysis

Human action analysis is a valued research area in computer vision because of its wide-ranging applications, including visual surveillance, human-computer interaction (HCI), motion analysis, and gaming. A typical process of vision-based human action analysis can be composed of four steps, namely human detection, human tracking, action recognition/retrieval, and high-level action evaluation [18]. Methodologies of action recognition/retrieval can be classified into single-layered approaches and hierarchical approaches [19]. Single-layered approaches analyze human actions of short movements based on a sequence of images with sequential characteristics, while the hierarchical approaches describe high-level actions as a combination of sub-movements and are suitable for analyzing human interactions. Among all single-layered approaches mentioned in [19], exemplar-based sequential approaches (which use DTW approach to deal with execution rate variation) can provide more flexibility and is able to cope with the problem of lacking training data.

Pose estimation is an important technique for action recognition/retrieval since actions can be modeled by the movements of body parts. Using color images captured by a single camera, Ramanan [20] applied conditional random field to represent the person as a pictorial structure composed of body parts tied together. Ferrari *et al.* [21] further integrated the pictorial structures among multiple frames with temporal and spatial information to reduce the search space and improve the estimation performance. The estimation results are then

applied to shot retrieval of movies either by querying on a single frame with the desired pose, or through a pose classifier trained from a set of pose examples [22]. Ferrari's method uses high-dimensional pose descriptors to represent either full probability distributions over possible part positions and orientations, or soft-segmentations of the parts, which is not practical for real-time retrieval task. In contrast, Jammalamadaka *et al.* [23] used a simple and effective representation based on a single absolute orientation of each part to achieve real-time pose retrieval. The recent hardware improvement of depth cameras has spurred the progress of human action analysis. People in top tier technical companies, such as HP and Microsoft, also endeavor to estimate and recognize human poses in real-time with the aid of the depth cameras (such as Kinect) [24], [25]. Shotton *et al.* [25] transformed the complex pose estimation problem into a simpler per-pixel classification problem. They trained very deep randomized decision forests using highly varied and large training dataset of depth images to achieve state-of-the-art recognition accuracy in super real-time.

Compared to human pose estimation, human action retrieval and human action recognition, there are extremely fewer works addressing the problem of high-level action evaluation, which is important for advanced applications such as the motion instruction systems used for learning dance moves or rehabilitation exercises. Raptis *et al.* [5] tried to classify dance moves in real-time and answer how well does the user perform the dance move. They used angular skeleton to represent the joint configurations, and then classified the input sequence into a defined dance move on the basis of a cascaded correlation-based classifier. After classification, DTW is used to form the distance metric between two dancing sequences and give a final score in terms of the overall performance. However, Raptis' work is not practical in general conditions since it highly relies on the assumption of musical beat alignment, and both the input and the compared video sequences should be roughly segmented in advance for calculating similarity. Alexiadis *et al*. [26] also focused on the evaluation of dance performance. They calculated three different scores, i.e., joint positions, joint velocities, and 3-D flow error, to wit the correctness of the action. The scores together with joint positions are presented in a virtual 3-D gaming environment, allowing the user to view and compare their movements with the teacher's from different viewpoints. Despite presenting an attractive visualization software, Alexiadis's system needs to process the data off-line. Thus, the user could only review the previously performed actions instead of getting real-time feedback while performing the movements.

Even though the complexity of the joint estimation process is reduced with the aid of the depth camera, finding representative feature descriptor of the posture, retrieving exact video segment of a given human action, and further evaluating the user's action performance in real-time with straightforward motion instructions are still noneasy tasks. Therefore, based on the skeleton estimation results obtained by the Kinect sensor and the OpenNI framework, we investigate into the problems of human movement retrieval and assessment to build a real-time human action tutor system.

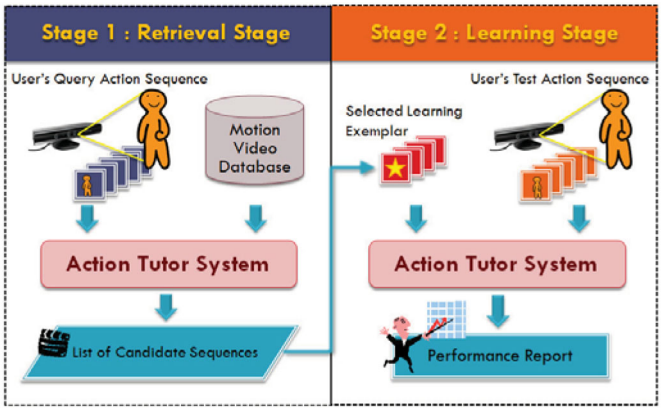

Fig. 1. From the user's perspective, the system can be divided into two stages, namely the retrieval stage, which searches target videos using the input action sequence, and the learning stage, which enables exercise learning by analyzing the user's movements based on the selected learning exemplar.

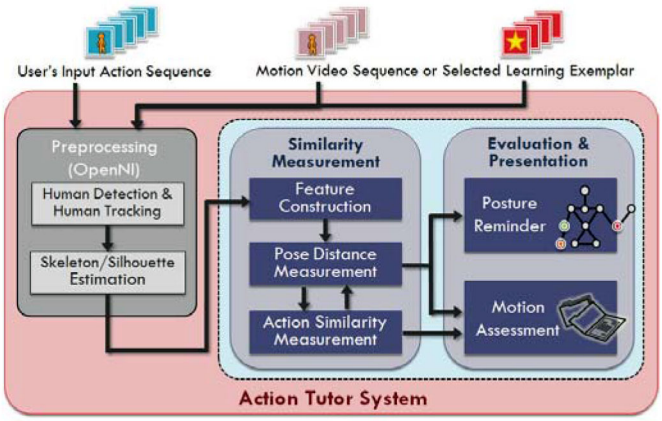

Fig. 2. Technical modules of the proposed action tutor system.

III. System Framework

We outline the system framework from two different perspectives. First, we introduce the system from the user's point of view by describing the input/output and the operation procedures. Then, we turn to the system internal and characterize functions of the core computational modules.

A. User's Perspective

From the user's point of view, the proposed system is operated through two stages as illustrated in Fig. 1. The first stage is the retrieval stage. In most of the existing motion assessment or instruction systems, before starting learning a specific motion, the user has to manually search the entire video database for the target movement. As the database grows abundantly, this search task gets more tedious and finally causes the user much burden. In contrast, using our system, the user could retrieve the movement they want to practice by simply performing it. To be precise, a target movement is roughly performed by the user and captured by Kinect or other 3-D sensors, resulting in a query action sequence. The system then takes this query sequence as input to search for similar video clips in the motion video database (each element in the

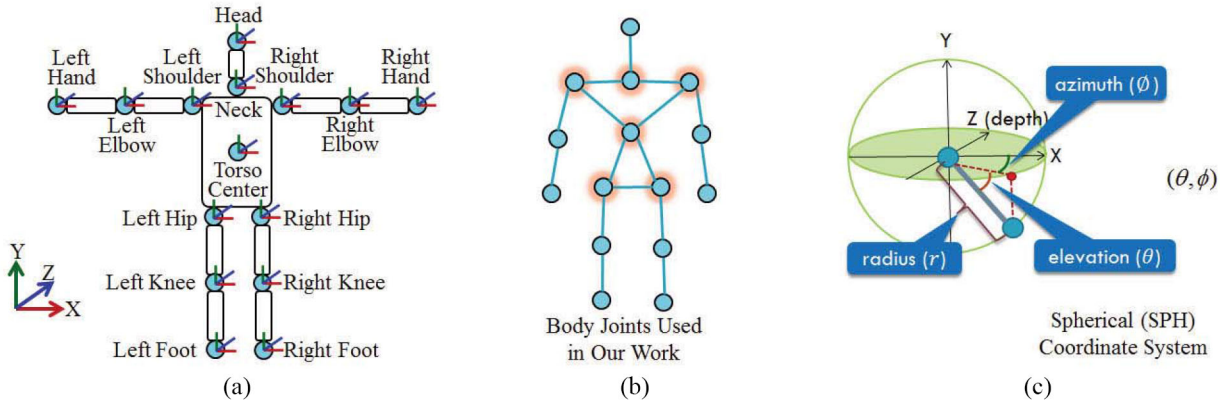

Fig. 3. (a) Coordinate system and human joint definitions in the OpenNI framework (for a user facing the sensor). (b) Fifteen joints used for movement analysis. Note that only the joints highlighted by a red halo are utilized to calculate the PCA of the torso direction. (c) Spherical coordinate system.

database is also a human action video captured by Kinect) and returns a ranked list consisting of segmented candidate sequences for the user to select the exact learning exemplar. The second stage is the learning stage. After choosing the exact learning exemplar, the user could follow and imitate it with real-time feedback pointing out body joints which are not posed correctly. The system gives the user motion instruction to correct the most inaccurate body joint. Moreover, the system can automatically play back the learning exemplar while the user fails to follow the movements. A detailed performance report will be presented in the end of the learning stage.

B. System's Perspective

Fig. 2 introduces the proposed action tutor system in terms of the three technical modules, namely preprocessing, similarity measurement, and evaluation/presentation modules. The system takes user's action sequence captured by Kinect (or other 3-D sensors) as input. The preprocessing module directly adopts the joint-position prediction and human silhouette extraction algorithms proposed in the OpenNI framework [6], [7] to obtain a sequence of joint-matched skeletons and human silhouettes. The skeleton/silhouette estimation technique is not the main focus of this paper and can be replaced by other estimation algorithms. The similarity measurement module then extracts representative features from the joint-matched skeleton sequence and the human silhouette sequence to measure the pose distance and action similarity between the user's query and each video in the motion video database (at the retrieval stage) or between the test action and the selected learning exemplar (at the learning stage). In the evaluation and presentation module, the pose distance results are integrated into an intuitive visualization to suggest the user how to adjust body parts that are not correctly posed, and the action similarity is output as a dynamic assessment of the user's action. The technical details of the system components will be described in Section V.

IV. Preprocessing

With the aid of Kinect and the full body analysis middleware developed in the OpenNI framework [6], [7], we can obtain the result of human pose estimation represented by the joint-matched skeleton and human silhouette in real time. The skeleton is composed of fifteen joints as illustrated in Fig. 3(a), and the position/orientation of each joint is estimated with a corresponding confidence score. These joint positions are given in the real world coordinates and measured in mm with $+X$ pointing to the right, $+Y$ pointing up, and $+Z$ pointing to the direction with increasing depth with respect to the Kinect sensor. However, the estimation results of some joints are not reliable and are often predicted with zero confidence. For example, the leg tracking is unstable and noisy unless the user stands with legs separated. Therefore, in this paper we propose two strategies to compensate the wrongly estimated joint positions. First, we apply weighted pose similarity measurement to decrease the similarity score of wrongly estimated joints (see Section V-C). Moreover, we also extract features from human silhouette to provide more accurate description of a human pose. Notice that the skeleton/silhouette estimation module can be replaced by other robust estimation algorithm. Considering the trade-off between estimation accuracy and real-time interaction, we choose the OpenNI framework to acquire skeleton/silhouette information for supporting the following analyses.

V. Similarity Measurement and Movement Evaluation

At the retrieval stage, the system has to not only search for videos containing the target action movement but also locate the time duration where the target action movement exactly occurs. According to the skeleton model and human silhouette obtained by the OpenNI framework, we construct representative features for describing human postures, and then define the distance between two static poses. Further, the technique of nonlinear time warping is utilized to calculate the similarity between two dynamic action videos, and the concept of substring finding is applied to identify the start/end time locations of the target action movement.

A. Skeleton Feature Construction

Given the skeleton model of a pose, we consider the local joint features and the global torso features to more appropriately describe a pose.

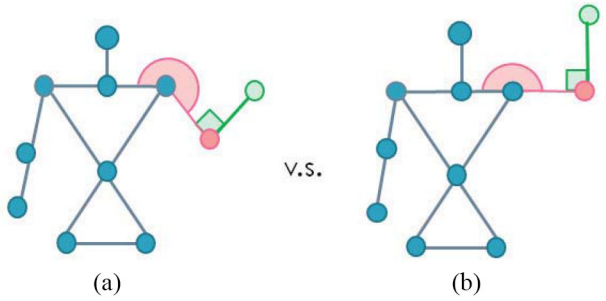

Fig. 4. Example of posture adjustment. (b) Incorrect posture can be adjusted to fit (a) target posture by only rotating the right shoulder.

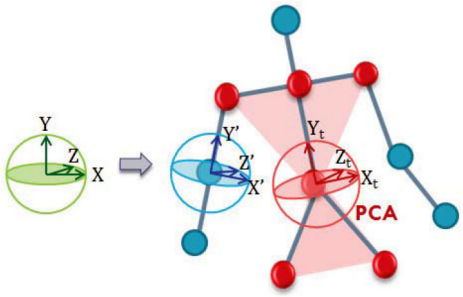

Fig. 5. Original Cartesian coordinate system with axes $[X, Y, Z]$ is rotated to the object-view coordinate system with axes $[X', Y', Z']$. The torso direction is represented by the principle components $[X_t, Y_t, Z_t]$ obtained from the five red joints related to the torso.

1) Local Joint Features: There are several aspects that need to be considered when representing the local position for each joint. First, the feature descriptor should not be affected by static transformations (i.e., the viewpoint transformations and the anthropometric transformations). Raptis *et al.* [5] have shown that using relative positions between joints instead of using absolute positions originating at the sensor is less dependent on the viewpoint. As shown in Fig. 3(b), the articulated skeleton model of the body can be taken as a tree rooted at Neck with other body parts radiating from it, and each joint can be represented by the relative position with respect to its parent joint. Moreover, the domain knowledge of biomechanics indicates that all diarthroses (movable joints) are synovial joints, which can be categorized into six different types, i.e., planar joint, hinge joint, pivot joint, condyloid joint, saddle joint, and ball-and-socket joint [27]. For example, the shoulder joint can be modeled as the ball-and-socket joint in which the distal bone is allowed to move around almost all directions. Even though we can use translations and rotations based on the X-, Y-, Z-axes (in a total of six degrees of freedom) to describe the relative relationship between two joints, the relative translations are usually much smaller than the relative rotations. Thus, the relative translations are ignored to simplify the calculation and we focus on the discussion of relative rotations.

The relative rotation among two joints can be calculated in the spherical (SPH) coordinate system, which transforms the Cartesian coordinates (x, y, z) into (r, θ, ϕ). Taking Fig. 3(c) as an example, using Right Shoulder as the reference point of Right Elbow, we could calculate the elevation θ with respect to the X-Z plane $(-(\pi/2) \leq \theta \leq (\pi/2))$, the azimuth ϕ with respect to the +X direction $(-\pi \leq \phi \leq \pi)$, and the radius r between Right Elbow and Right Shoulder. Note that the radius element r is not considered in this paper because each skeleton has to be normalized so that the representation would not be affected by human anthropometric ratios or the distance between the user and the camera. Using the SPH representation, i.e., (θ, ϕ), the issue of anthropometric transformations can be solved easily.

The second issue we have to tackle is illustrated in Fig. 4. For two postures A and B, we may think the posture B is incorrect because the Right Elbow joint and the Right Hand joint are improperly posed compared with the target posture A. However, the posture B can be easily adjusted to fit the target posture A by only rotating the right humerus and keep other joints fixed in the same relative positions. This kind of relationship cannot be captured by the SPH representation transformed directly from the original Cartesian coordinates, i.e., (x, y, z). A better choice would be using the object-view to describe the joint positions. That is, before transforming to the SPH coordinate system, the Cartesian coordinate axes of each child-joint should be rotated by aligning the zenith direction with the correspondingly preceding proximal bone. Fig. 5 shows an example: using the Left Elbow joint as the reference point, the axes of the original Cartesian coordinate system $[X, Y, Z]$ is first aligned with the left humerus, and the new axes of the object-view coordinate system are notated as $[X', Y', Z']$. Then, the Left Hand joint can be represented by the elevation θ' and the azimuth ϕ' calculated with respect to the X'–Z' plane and $+X'$ direction, respectively.

2) Global Torso Features: To more completely describe the global characteristic of a given pose skeleton, we preserve the torso direction information modeled by five joints related to the torso, i.e., torso center, left/right shoulder, and left/right hip joints. Principal component analysis (PCA) is applied to these five joints in the original Cartesian coordinate system with axes $[X, Y, Z]$, and the obtained principal components are used to be the three axes of the torso direction [5], i.e., $[X_t, Y_t, Z_t]$, as shown in Fig. 5. Each axis is a 3-D vector, resulting in a nine-dimensional feature vector for describing the axes of a global torso direction. To reduce the feature dimension but still keep the same information, we represent the torso direction by three Euler angles (α, β, γ), which sequentially rotates the original coordinate system with respect to the three axes $[X, Y, Z]$ [28], that is

$$R(\alpha, \beta, \gamma) = R_Z(\alpha) R_Y(\beta) R_X(\gamma). \tag{1}$$

Moreover, the relation between the original Cartesian coordinate system and the torso direction can be represented by

$$[R][X\ Y\ Z] = [X'\ Y'\ Z'] \tag{2}$$

which means the rotation matrix R can be calculated as

$$[R] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [X'\ Y'\ Z'][X\ Y\ Z]^{-1}. \tag{3}$$

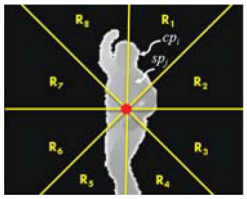

Fig. 6. Illustration of how to construct silhouette features.

The Euler angles (α, β, γ) can then be solved by [28]

$$\alpha = \tan^{-1}(r_{21}/r_{11}) \tag{4}$$

$$\beta = \tan^{-1}\left(-r_{31}\Big/\sqrt{r_{32}^2 + r_{33}^2}\right) \tag{5}$$

$$\gamma = \tan^{-1}(r_{32}/r_{33}). \tag{6}$$

The range of the arctangent can be extended to $[0, 2\pi)$ by the function provided in the C programming language.

B. Silhouette Feature Construction

We take the shape and depth distribution of human silhouette into consideration to describe a human pose since joints estimated by Kinect are not always reliable (especially when the user's joints are occluded by other body parts). Given a human silhouette, we first find the centroid of all contour points and divide the frame into $|R|$ ($|R| = 8$ in our work) regions by radiate lines emitting from the centroid (as illustrated in Fig. 6). The shape and depth features of the human pose are then extracted as follows.

1) Shape Features: For each region R_r, we calculate two descriptors to depict the shape characteristic of the human silhouette in R_r. That is

$$HS_{r,1} = mean\,(dis(centroid, cp_i)) \tag{7}$$

$$HS_{r,2} = var\,(dis(centroid, cp_i)) \tag{8}$$

where $dis(centroid, cp_i)$ is the 2-D Euclidean distance between the centroid and each contour point in R_r.

2) Depth Features: For each region R_r, we also calculate two descriptors to depict the depth distribution of the human silhouette in R_r. That is

$$HS_{r,3} = mean\left(depth(sp_j)\right) \tag{9}$$

$$HS_{r,4} = var\left(depth(sp_j)\right) \tag{10}$$

where $depth(sp_j)$ is the relative depth between the centroid and each silhouette point in R_r.

C. Pose Distance Measurement

Given two pose A and B, the distance between them can be defined by

$$D_{pose}(A, B) = \frac{w_{\text{skeleton}}}{1 + \sum_{k=1}^{K} w_k}\left[\Delta TD(A, B) + \sum_{k=1}^{K} w_k \Delta J_i(A, B)\right] + w_{\text{silhouette}}\,\Delta HS(A, B) \tag{11}$$

where $\Delta TD(A, B)$ is the global skeleton difference defined by the torso directions of A and B, $\Delta J_i(A, B)$ is the local skeleton difference at the ith joint between A and B (K is the total number of the considered local joints), and $\Delta HS(A, B)$ is the human silhouette difference. w_{skeleton} and $w_{\text{silhouette}}$ control the importance of skeleton and silhouette information, respectively. We set both of them to be 0.5 in our work. w_k weights the influence of the local difference on each corresponding joint, which can be set according to the application. In our work, since the OpenNI framework extracts unstable and noisy leg-joint positions, a smaller w_k is assigned to each lower-body joint and a larger w_k is assigned to each upper-body joint.

The SPH representations obtained from the object-view coordinates are utilized to calculate each local joint difference, that is

$$\Delta J_i(A, B) = \frac{1}{2}\left[\Delta\theta_i'(A, B) + \Delta\phi_i'(A, B)\right] \tag{12}$$

$$\Delta\theta_i'(A, B) = \frac{1}{\pi}\left|\theta_i'(A) - \theta_i'(B)\right| \tag{13}$$

$$\Delta\phi_i'(A, B) = \frac{1}{\pi}\min\left(\left|\phi_i'(A) - \phi_i'(B)\right|, 2\pi - \left|\phi_i'(A) - \phi_i'(B)\right|\right) \tag{14}$$

where $0 \le \Delta\theta_i'(A, B), \Delta\phi_i'(A, B) \le 1$. Moreover, the global torso direction difference is calculated by

$$\Delta TD(A, B) = \frac{1}{3}\left[\Delta\alpha(A, B) + \Delta\beta(A, B) + \Delta\gamma(A, B)\right] \tag{15}$$

where the $\Delta\alpha(A, B)$, $\Delta\beta(A, B)$, and $\Delta\gamma(A, B)$ are calculated in the similar way as $\Delta\phi_i'(A, B)$. Since the joint estimation results are not always reliable for the two Shoulder joints due to occlusion. Therefore, we take two Hip joints into account to more accurately find the torso direction and compensate the erroneously measured pose distance resulting from the local joint difference. The human silhouette distance is defined by

$$\Delta HS(A, B) = \frac{1}{4|R|}\sum_{r=1}^{|R|}\sum_{d=1}^{4}\Delta HS_{r,d}(A, B) \tag{16}$$

$$\Delta HS_{r,d}(A, B) = \frac{1}{d_{\max}}\left|HS_{r,d}(A) - HS_{r,d}(B)\right| \tag{17}$$

where $d_{\max}$ is a sufficiently large distance trained by our dataset.

D. Action Similarity Measurement

Given a query action sequence Q and a compared action video V in the database (the video V may be much longer than the query Q and may be composed of multiple actions), we measure the difference between them and find the exact time duration where the query action Q occurs in the video V. Three methods including cross-correlation, DTW, and ASM are investigated in our work, and we propose a modified-DTW method to meet the demands of our system.

1) Cross-Correlation: In the field of signal processing, cross-correlation is commonly used to measure the similarity and capture the shape alikeness between two waveforms. It could be used to search for a shorter template signal in a long signal by applying a time delay to the template signal. We utilize this correlation-based method to locate the query action Q in the video sequence V. Both Q and V are first normalized to

lower the influence caused by the prior distribution (standard deviation) of each element in the feature vector representing a pose skeleton. For the ith element of the feature vector, the normalized cross-correlation (NCC) score between Q and a subsequence S (which has the same length with Q) of V can be written as

$$\text{corr}[i] = \frac{\sum_{t=1}^{N} Q_i(t)S_i(t) - N\bar{Q}_i\bar{S}_i}{\sqrt{\sum_{t=1}^{N} Q_i(t)^2 - N\bar{Q}_i^2}\sqrt{\sum_{t=1}^{N} S_i(t)^2 - N\bar{S}_i^2}} \tag{18}$$

where $\bar{Q}_i$ and $\bar{S}_i$ are the means of signals Q_i and S_i, and N is the length of the query sequence Q.

Considering all elements in the L-dimensional feature vector describing a pose, the overall correlation score between the two sequences Q and S is

$$\text{corr}^* = \frac{1}{L}\sum_{i=1}^{L} \text{corr}[i]. \tag{19}$$

A threshold Th_{corr} is used to detect high correlated subsequence S in V with respect to Q. However, directly comparing two sequences frame-by-frame will suffer from the issue of temporal transformations. Although we can simply apply linear time-warping methods, such as up-sampling or down-sampling beforehand, the correlation-based method still fails when the two action sequences Q and S vary in not only the overall execution duration but also the local execution speed (caused by acceleration and/or deceleration during performing the action). In contrast, nonlinear time-warping methods allowing similar signal shapes to be matched even if they are out of phase in the time axis would more appropriately measure the similarity between two action sequences.

2) Dynamic Time Warping: DTW is a nonlinear time-warping scheme which aims to find the best warping function between two input signals with minimum total distance under certain constraints. It could determine the similarity measurement with tolerance to a certain degree of time variation between sequences. Several constraints are often used to reduce the search space and set a reasonable tolerance range. For example, the monotonicity constraint prevents the warping path from going back in time axis, and the boundary conditions limit the warping path to start from the first time instance and end at the last time instance for both the query sequence Q and the compared sequence V. The DTW problem is often solved by the divide-and-conquer approach and implemented using the technique of dynamic programming. As illustrated in Fig. 7(a), given the query sequence Q composed of successive poses $\{q_1, q_2, \ldots, q_M\}$ and the compared video sequence V containing successive poses $\{v_1, v_2, \ldots, v_N\}$, a DTW table of size $M \times N$ is created and its boundaries are set as infinity. For $1 \le i \le M$ and $1 \le j \le N$, each grid (i, j) is filled with a minimum warping distance defined by

$$d_w(i,j) = \min\begin{pmatrix} d_w(i-1,j-1) \\ d_w(i,j-1) \\ d_w(i-1,j) \end{pmatrix} + \text{cost}(i,j) \tag{20}$$

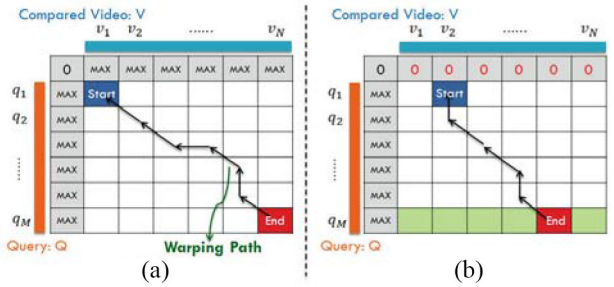

Fig. 7. DTW with different initializations of the boundary conditions and various strategies of backtracking. (a) DTW. (b) Modified DTW.

where $\text{cost}(i, j)$ is the difference between poses q_i and v_j, i.e., $D_{\text{pose}}(q_i, v_j)$ defined in Section V-C. After filling the entire table, the distance between Q and V are defined by $d_w(M, N)$. The conventional DTW method then backtracks from the end grid (M, N) to the start grid $(1, 1)$ and reconstruct the entire alignment path. With the detailed alignment between the two sequences, the DTW-based similarity measurement is invariant to the temporal transformation mentioned in Section I.

3) Approximate String Matching: ASM (also called fuzzy string searching) is a special case of DTW and is originally used to solve the string matching problem. Given the query string Q composed of pose symbols $q_1, q_2, \ldots, q_M$ and the compared string V composed of pose symbols $v_1, v_2, \ldots, v_N$, ASM tries to find a substring S of V which has the smallest edit distance to transform V into Q. In other words, ASM can find the occurrence of the query pattern Q in a long sequence V by calculating the minimum edit distance between Q and V. More precisely, let $d_e(i, j)$ denote the minimum edit distance to transform the first j symbols of V into the first i symbols of Q. At each symbol v_j, the editing operations are as follows.

1) *Match or Substitution:* The pose symbol v_j is matched with pose symbol q_i or is substituted by pose symbol q_i with an additional cost $\delta(q_i, v_j)$.
2) *Insertion:* There is an extra pose symbol q_i in Q to be considered, and we have to insert q_i into V (which is equivalent to deleting the symbol q_i from Q) with an additional insertion cost $\delta(\epsilon, q_i)$.
3) *Deletion:* There is an extra pose symbol v_j in V to be considered, and we have to delete v_j from V with an additional deletion cost $\delta(v_j, \epsilon)$.

ASM can be also solved by dynamic programming, and the total edit distance at grid (i, j) is defined by

$$d_e(i,j) = \min\begin{pmatrix} d_e(i-1,j-1) + \delta(q_i, v_j) \\ d_e(i,j-1) + \delta(\epsilon, q_i) \\ d_e(i-1,j) + \delta(v_j, \epsilon) \end{pmatrix}. \tag{21}$$

The match/substitution cost $\delta(q_i, v_j)$ is determined by the pose difference $D_{\text{pose}}(q_i, v_j)$ defined in Section V-C. Moreover, the cost of inserting or deleting a pose symbol s is set to be a constant value $C = 0.5$, which is the distance between pose s and an empty pose ϵ.

With different initializations of the boundary conditions and various ways of backtracking, ASM can be applied to two kinds of matching, the entire string matching and the multiple substring matching. As illustrated in Fig. 8(a), if we want

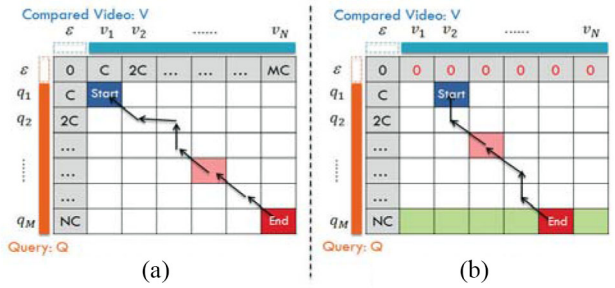

Fig. 8. ASM with different initializations of the boundary conditions and various strategies of backtracking. (a) Entire string matching. (b) Substring matching.

to match the query sequence Q against the entire string of sequence V, the cost $d_e(i, 0)$ (i.e., the cost of deleting the first i symbols from Q) is set to be $i \cdot C$. Similarly, the cost $d_e(0, j)$ is set to be $j \cdot C$. Moreover, the warping path of the entire string matching is constrained by the backtracking path from grid (M, N) to grid $(1, 1)$. On the other hand, if we want to find the duration where Q exactly occurs in V, the problem is transformed to a substring matching problem, which means Q may start from any place in V and there should be no penalty for deleting the first j symbols from V. As shown in Fig. 8(b), the cost $d_e(0, j)$ is set to be zero, and the start/end points of the backtracking path for substring matching are not constrained. That is, we have to search the entire last row for the minimum editing distance to find the best match result.

4) Modified DTW: Our system aims to find all video segments (with exact start/end time) having movements similar to the query. However, conventional DTW have two main problems. First, the cost can only find one warping path between Q and V. If the query movement Q appears more than once in a long video V, we can only find one of them since the backtracking step is simply applied once to trace the path starting from the grid (M, N). Second, if the query movement Q appears near the start of a video V_1 and also appears near the end of a video V_2, the one in V_2 might have much smaller similarity score than the one in V_1 even though these two video segments are the same. That is because the boundary condition $d(0, j)$ given in conventional DTW are all infinity. Therefore, we propose a modified DTW method, which keeps the distance function the same as the one in conventional DTW but utilizes the boundary conditions/backtracking strategies as the ones used for the ASM-based substring matching.

The major difference between DTW-based methods and ASM-based methods is their points of view toward the insertion and the deletion operations. Since ASM aims to solve the string editing problem, when an insertion or a deletion occurs, the distance function is set to be the distance of the preceding path plus the difference between the current symbol and an empty symbol. That is to say, it treats the extra symbol as noise in essence and a fixed cost is adopted to lower the effect caused by the noise. In contrast, the DTW method sets the distance function to be the preceding distance plus the difference between the extra symbol and the current symbol. It can be viewed as applying up- or down-sampling locally to compensate for the missing information when insertion/deletion occurs. In Section VI-B, we will compare the performance between the substring matching method based on ASM and the proposed modified DTW approach. At the retrieval stage of our action tutor system, the user can choose the cross-correlation method, substring matching version of ASM, or the modified DTW to find similar video segments in the database. Once the user chooses an exemplar video segment in the learning stage, the entire string matching based on ASM or the conventional DTW can be selected to evaluate whether the user successfully follows the movements of the learning exemplar.

E. Movement Evaluation and Presentation

As shown in the demo video,[1] to clearly instruct the user how to adjust his/her postures at the learning stage, we indicate wrongly posed joints with red circles and display the rest with green circles. The joint difference defined in (12) is used to evaluate the correctness of each posed joint. Moreover, the joint with the largest joint difference is accompanied with a blue arrow pointing to the direction where it should be adjusted to. The overall posture difference and the dynamic action similarity at each time instance are also presented, where the action similarity at the time instance i is calculated by applying DTW to the video sequences composed of the corresponding past N frames. When the dynamic action similarity at the time instance i is smaller than TH_{sim}, the system will automatically stop playing the learning exemplar and play back from the $(i-N)$th frame. The playing back mechanism prevents the user from skipping learning important movements while trying to follow the learning exemplar.

VI. Experiment Results

We conducted objective and subjective tests to evaluate the performance of our system. First, we investigate what kind of skeleton feature is the most appropriate choice for pose similarity measurement. We then compare different nonlinear time warping methods for action similarity measurement. Finally, we report the results of the subjective experiments to evaluate the overall performance of our action tutor system. An action dataset[2] containing movement videos of ten different actions is collected for our evaluation. These ten actions are: 1) level hand swing (LHS); 2) elevated hand swing (EHS); 3) arm rotation (AR); 4) left–right elevated hand swing (LR-EHS); 5) golf-swing; 6) rod-swing; 7) pitch; 8) tai-chi; 9) jack; and 10) kick. Among these actions, 1)–4) are upper-body movements of the Chinese Qigong exercise promoted by Meimen Qigong Culture Center,[3] and 5)–10) are actions with lower-body movements. Moreover, some of these actions includes occluded joints and fast movements (please refer to the action dataset website). Each action was performed by 11 individuals (7 males and 4 females) and each individual performs the same action twice. The first performance contains only the assigned action movement, while the second performance involves other

[1]Demo video of the proposed Action Tutor system: http://www.youtube.com/TSMCActionTutor

[2]Action Dataset: http://www.cmlab.csie.ntu.edu.tw/~trimy/MovementAssessment/HMRA.html

[3]Meimen Qigong Culture Center: http://www.mymeimen.org/

TABLE I
NOTATIONS, CORRESPONDING FACTOR ATTRIBUTES, AND DIMENSIONS OF DIFFERENT SKELETON DESCRIPTORS

	Axes Rotation	Torso Direction	Joint Selection	Dim.
SPH [5]	X	X	X	16
SPH-S	X	X	O	8
R-SPH [5]	O	X	X	16
R-SPH-S	O	X	O	8
R-SPH+T	O	O	X	19
R-SPH-S+T	O	O	O	11

movements before or after the assigned action movement. Both the color and the depth information is recorded/encoded by Kinect and OpenNI with the resolution of 640 × 480 and the default frame rate of 30 frames/s.

A. Evaluation of Features

In Section V-A, we define skeleton-based features to distinguish actions in and out of the class. Here we further investigate three different factors that might influence the effectiveness of the skeleton feature.

1) Whether axes rotation is applied to local joint features, i.e., X, Y, Z axes are rotated to X', Y', Z' or not.
2) Whether global torso direction is considered.
3) Whether joint selection is adopted.

Take the LHS action as an example, since this action mainly focus on the movements of arms, we can select only Left/Right Elbow and Left/Right Hand joints to represent the local joint features [i.e., in (11), $w'_k s$ for the Left/Right Elbow and Left/Right Hand joints are set to be one and $w'_k s$ for other joints are set to be zero] or keep the information of all body joints (i.e., $w'_k s$ for all joints are set to be one). The notations, corresponding factor attributes, and dimensions of different skeleton feature descriptors are listed in Table I. Note that the position of the root joint, i.e., Neck, is only used as the reference point for other joints and is not included in the descriptors.

We apply NCC to the query action Q and a video sequence V, and the correlation responses along the time axis are used to visualize the performance of each skeleton-based descriptor. Since we aim to investigate factors related to skeleton, silhouette features are not considered in calculating NCC in this experiment. We take the LHS action as an example to compare the results. LHS is an action performed by standing with feet shoulder-width apart, raising both arms to chest height with arms parallel to each other and parallel to the ground, and then swinging both arms back and forth five times with the same speed. At the fifth swing, the performer bends the knees and dip down twice. Therefore, a complete cycle of the LHS action is composed of five sub-movements and the movement structure can represented by $AAAAA'$, where the fifth sub-movement A' is slightly different from the former four. The user is asked to perform the query Q, which only contains a sub-movement A, and then Q is compared with another long video sequence V involving the LHS action performed by the experienced student of Meimen.

Fig. 9(a) shows the cross-correlation responses obtained by using different kinds of skeleton-based descriptors when Q and V are performed by the same player with similar execution duration and captured from similar viewpoints. Since the cross-correlation calculates the response of the current tth frame using the feature signal information in the past N frames (N is the duration of the query Q), we shift forward the response curve by N frames to align the response peaks with the starting points where similar signal patterns occur. The ground truth of a complete movement A or A' is indicated by the interval between two successive dash lines, and the black dash lines denotes the end of the movement A'. In this case, response peaks for all kinds of descriptors can be clearly detected at the start time of the movement ground truth, which means the performance of different kinds of descriptors are equally well. However, as shown in Fig. 9(b), when Q and V are performed by different people with different execution duration and captured from slightly different viewpoints (i.e., with anthropometric/temporal/viewpoint transformations), feature descriptors such as SPH and SPH-S will probably get high responses at the time outside the time periods in which the movement A or A' actually occurs. As shown in Fig. 9, even though the basic SPH and SPH-S descriptors perform well with correlation response up to 0.8 when movements are performed by the same actor, we could not use them as the representation for action similarity measurement when different people perform them because the correlation response in the in-class region is not exactly larger than that in the out-of-class region. In contrast, descriptors with axes rotation have more consistent performance. For example, although the maximum response using R-SPH-S is only around 0.6 in Fig. 9(b), the response in the in-class region is exactly larger than that in the out-of-class region, which means the R-SPH-S feature would be suitable for evaluating movement similarity.

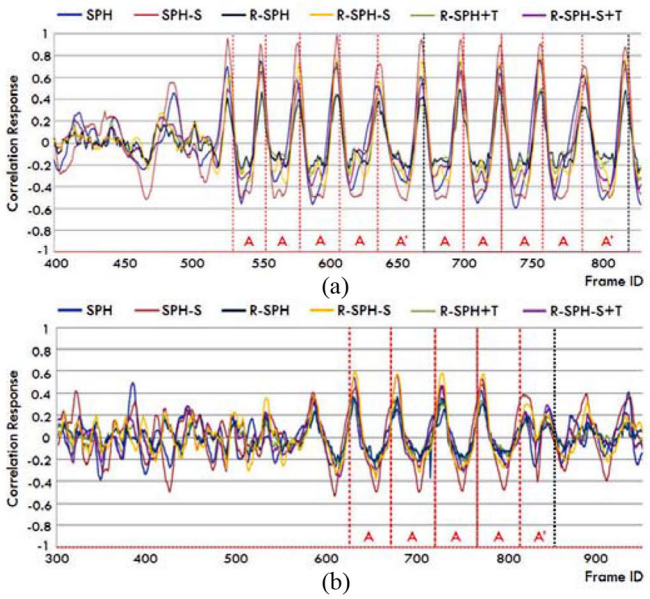

Fig. 9. Cross-correlation responses between Q and V obtained by using different kinds of feature descriptors. (a) Q and V are performed by the same player with similar execution duration and captured from similar viewpoints. (b) Q and V are performed by different people with different execution duration and captured from slightly different viewpoints.

In the case of the LHS action, we observed that applying joint selection would not affect the shape of the response signal

but would reflect higher contrast in the amplitude and make the response signal smoother. That is to say, joint selection helps us to more easily detect the time positions where the query movements occur in the video V. This may be caused by the noisy nature of the sensor and so are the estimated joint positions. The removed joints are relatively static; therefore, feature descriptors involving them are more easily affected by the jittering data. Moreover, among feature descriptors with joint selection, the performances of R-SPH-S and R-SPH-S+T are much better than SPH-S.

Another issue is how to determine the degree of static (i.e., how to determine w_k) for each joint of an action. In this paper, given a query video, we assign w_k a larger value if the variances of ϕ_i' and θ_i' for the joint i are larger. We also observed that applying R-SPH-S+T would benefit the detection for action movements with torso rotations, while R-SPH-S performs better when the query action does not involve much change in torso rotation. Therefore, variance of the torso direction, i.e., $var(TD)$, for a query is calculated and we apply R-SPH-S+T to measure the pose distance if $var(TD)$ is larger than a predefined threshold; otherwise, R-SPH-S is applied.

In addition to skeleton features, we also consider silhouette features to represent a human pose. We compare the proposed features with the descriptors employed in [5] (which only considers skeleton information) and [29] (which only utilized depth information inside the human silhouette). The retrieval accuracies of using skeleton-only [5], silhouette-only [29], and our skeleton-silhouette descriptors are evaluated based on the collected action database. Each video containing only the assigned action movement is used to be the query video once, and the whole collected action database (except the query video) is used to be retrieved at the same time. Therefore, we have 11 queries for each action type. Given a query video Q_i, the modified DTW is applied to search the database, and the start/end time points of the top-20 video segments having the highest similarity scores are obtained. The average precision (AP) defined by

$$AP(Q_i) = \frac{\sum_{k=1}^{n} (P(k) \cdot hit(k))}{\text{number of hits}} \tag{22}$$

is used to evaluate the retrieval accuracy of the query Q_i, where k is the rank in the retrieved list, and n is number of the returned candidates. $hit(k)$ is an indicator function which equals 1 if the intersection duration of the estimated kth segment and the ground truth segment are longer than or equal to 50% of their union, and equals zero otherwise. $P(k)$ is the precision at cut-off k in the list. The retrieval performance for a set of queries is then evaluated by mean average precision (MAP) defined as

$$MAP = \frac{\sum_{i=1}^{H} AP(Q_i)}{H} \tag{23}$$

where H is the number of queries. Fig. 10 shows the corresponding MAP of each action category and the average MAP over all queries regardless of the associated action categories. Overall, the proposed descriptor that considers both skeleton and silhouette information performs better than the others.

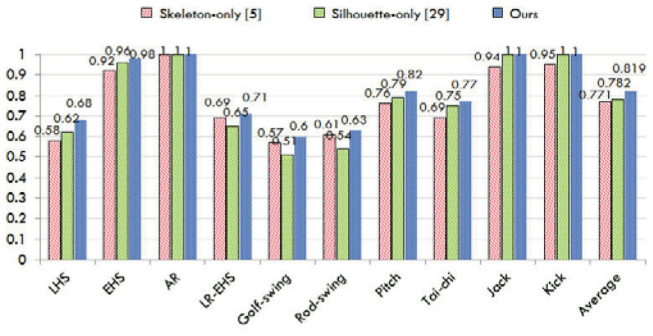

Fig. 10. MAP of each action category and the average MAP of all queries obtained by using modified DTW with different kinds of features.

B. Evaluation of Nonlinear Time Warping

As mentioned in Section V-D1, the correlation-based method highly depends on the length of the query action and therefore can not find the exact end points of the action occurrences unless we apply up/down-sampling to the query action and exhaustively execute the cross-correlation method using queries of different time lengths. Instead, nonlinear time warping methods are more effective to measure the difference between two action videos and are capable of efficiently locating the duration where the query Q exactly occurs in the video V. In this section, evaluation of different nonlinear time warping methods introduced in Section V-D are conducted on the collected action database. The conventional DTW and the entire string matching version of ASM cannot meet the demand at the retrieval stage of the proposed action tutor system since they require the boundaries of the compared videos to be aligned in advance. Therefore, we only apply the modified DTW and the substring matching version of ASM to the collected database.

Fig. 11 shows the corresponding MAP of each action category and the average MAP over all queries regardless of the associated action categories. We observed that the neither the modified-DTW NOR the substring matching version of ASM can always perform better than the other for all kinds of actions. Therefore, in the proposed Action Tutor system, we let the user to select the method to retrieve videos. The retrieval performance is also influenced by the incorrect joint estimation results obtained with the OpenNI framework, especially when the arms move to the back and are invisible for a while. In the future we will use multiple Kinect sensors to overcome this nature limitation of the single-view vision-based action analysis approach.

C. User Evaluation

We also conducted subjective experiments to evaluate the overall performance of our action tutor system. Eighteen users were invited to join the user study, including eleven males and seven females aged between thirteen to fifty years old. These participants were requested to complete the entire process of our system from the retrieval stage to the learning stage and then answered a questionnaire as listed in Table II. The participants are asked to give a satisfaction score ranging from 1 to 7 for each question, where 7 = strongly agree, 4 = neutral, and 1 = strongly disagree. The questionnaire aims to evaluate the proposed system in terms of three aspects, i.e.,

TABLE II
QUESTIONNAIRE OF THE SUBJECTIVE USER STUDY AND THE CORRESPONDING AVERAGE SCORES

Aspect	Quantitative Question	S_{mean}	S_{var}
Effectiveness	• Do you think the proposed system correctly find your target movements?	5.89	0.58
	• Compared to the "LET'S DANCE with Mel B" [1], do you think the proposed system provides more proper instructions to adjust your action movements?	5.89	0.46
	• Do you think the proposed system can effectively help you to learn exercise movements?	6.33	0.35
Efficiency	• Do you think the proposed system helps you quickly find the target movements in the motion video database?	6.17	0.74
	• Do you think the proposed action tutor system is convenient and easy to operate?	5.67	0.47
Acceptance	• Do you think learning exercise by using the proposed system is interesting?	6.39	0.25
	• Compared to learning exercise with a human tutor, do you think learning with the proposed action tutor system makes you feel less embarrassed?	6.17	0.26
	• Would you like to use the proposed system as a long-term tutor for learning exercise movements?	6.05	0.41

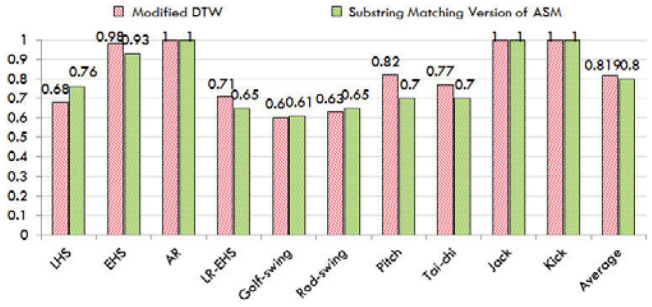

Fig. 11. MAP of each action category and the average MAP of all queries obtained by using modified DTW or substring matching version of ASM. The experiments are conducted on queries with/without presegmentation.

effectiveness, efficiency, and acceptance. Both the mean and variance of the score with respect to each question are reported in Table II, and the results show that the proposed system is effective for learning exercise movements and is efficient for finding target movements. Moreover, most users are willing to use the proposed system as a long-term tutor because it is interesting and will not bring embarrassment while learning movements compared to learning with a human instructor.

VII. CONCLUSION

We propose an action tutor system which achieves high-level evaluation of human action movements with the aid of Kinect. The system is operated in two stages: at the retrieval stage, the user can search the video database for the target action movements by different action matching methods. A list of video candidates are returned to the user for choosing the learning exemplar. At the learning stage, the user follows the movements in the learning exemplar, and the system evaluates the detailed pose difference and the accumulated action similarity between the user and the exemplar in real-time. We construct representative pose features based on both skeleton and silhouette information. Techniques of nonlinear time warping, i.e., modified DTW and the substring matching version of ASM, are applied to tackle the issue of temporal transformations while retrieving target videos, and experiments conducted on the videos of ten different actions show that the proposed features and matching methods are effective for movement retrieval. The subjective test also reveals that the proposed system is effective, efficient, and acceptable to be used for learning exercise movements.

Lots of related issues are worthy of further investigation, for example, improving the joint estimation algorithm to obtain more robust joint position information, using multiple Kinect sensors to capture multiview information of the human action movements, and integrating domain knowledge to give different penalty for each joint while calculating the pose difference. Also, we will experiment our system with a larger collection of testing videos with more diversity of action movements. Moreover, we will apply the proposed action tutor system to medical rehabilitation and game design in the near future.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Council of R.O.C. under grants NSC 102-2218-E-006-005.

REFERENCES

[1] Lightning Fish Studios. (2011, Jun.). *Let's Dance with Mel B* [Online]. Available: http://www.kinectaku.com/games/360/lets_dance_with_mel_b
[2] Harmonix Music Systems. (2011, Jun.). *Dance Central* [Online]. Available: http://en.wikipedia.org/wiki/Dance_Central
[3] Ubisoft. (2011, Oct.). *Just Dance* [Online]. Available: http://just-dance-thegame.ubi.com/jd-portal/en-gb/home/index.aspx
[4] Y. Sheikh, M. Sheikh, and M. Shah, "Exploring the space of a human action," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Washington, DC, USA, Oct. 2005, pp. 144–149.
[5] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animat. (SCA)*, New York, NY, USA, 2011, pp. 147–156.
[6] OpenNI Organization. (2011, Jan. 19). *OpenNI User Guide* [Online]. Available: http://www.openni.org/documentation
[7] PrimeSense Inc. (2011, Jan. 19). *Prime Sensor NITE 1.3 Algorithms Notes* [Online]. Available: http://www.primesense.com668
[8] L. Cruz, D. Lucio, and L. Velho, "Kinect and RGBD images: Challenges and applications," in *Proc. SIBGRAPI Conf. Graph. Patterns Images Tuts. (SIBGRAPI-T)*, Ouro Preto, Brazil, 2012, pp. 36–49.
[9] B. Packer, K. Saenko, and D. Koller, "A combined pose, object, and feature model for action understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1378–1385.
[10] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2834–2841.
[11] H.-H. Hsu, Y. Chiou, Y.-R. Chen, and T. K. Shih, "Using kinect to develop a smart meeting room," in *Proc. Int. Conf. Netw.-Based Inf. Syst.*, Gwangju, Korea, 2013, pp. 410–415.
[12] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2759–2766.
[13] S. Liu *et al.*, "Video stabilization with a depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, 2012, pp. 89–95.

[14] F. Yang, J. Huang, X. Yu, X. Cui, and D. Metaxas, "Robust face tracking with a consumer depth camera," in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, USA, 2012, pp. 561–564.
[15] W. Shen *et al.*, "Exemplar-based human action pose correction and tagging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1784–1791.
[16] J. Shen and S.-C. Cheung, "Layer depth denoising and completion for structured-light RGB-D cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 1187–1194.
[17] J. Barron and J. Malik, "Intrinsic scene properties from a single RGB-D image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 17–24.
[18] L. Chen and C. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *Int. J. Web Inf. Syst.*, vol. 5, no. 4, pp. 410–430, 2009.
[19] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16.1–16.43, Apr. 2011.
[20] D. Ramanan, "Learning to parse images of articulated bodies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006.
[21] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.
[22] V. Ferrari, M. Marin Jimenez, and A. Zisserman, "Pose search: Retrieving people using their pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1–8.
[23] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar, "Video retrieval by mimicking poses," in *Proc. ACM Int. Conf. Multimedia Retrieval*, Hong Kong, China, 2012, p. 34.
[24] H. P. Jain, A. Subramanian, S. Das, and A. Mittal, "Real-time upper-body human pose estimation using a depth camera," in *Proc. 5th Int. Conf. Comput. Vis./Comput. Graph. Collaboration Tech.*, Rocquencourt, France, 2011, pp. 227–238.
[25] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jun. 2011, pp. 1297–1304.
[26] D. S. Alexiadis *et al.*, "Evaluating a dancer's performance using kinect-based skeleton tracking," in *Proc. 19th ACM Int. Conf. Multimedia*, New York, NY, USA, 2011, pp. 659–662.
[27] E. Marieb, *Essentials of Human Anatomy & Physiology*. Redwood City, CA, USA: Pearson/Benjamin Cummings, 2011.
[28] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2006.
[29] Y.-C. Lin, M.-C. Hu, W.-H. Cheng, Y.-H. Hsieh, and H.-M. Chen, "Human action recognition and retrieval using sole depth information," in *Proc. 20th ACM Int. Conf. Multimedia*, New York, NY, USA, 2012, pp. 1053–1056.

Min-Chun Hu (M'14) is also known as Min-Chun Tien and Ming-Chun Tien. She received the B.S. and M.S. degrees in computer science and information engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, in 2011.

She is an Assistant Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. She was a Post-Doctoral Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, from 2011 to 2012. Her current research interests include digital signal processing, digital content analysis, pattern recognition, computer vision, and multimedia information system.

Chi-Wen Chen received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2009 and 2012, respectively.

She was a Research Assistant with Academia Sinica, Research Center for Information Technology Innovation, Taipei, Taiwan, and has been an Application Innovation Consultant with IBM, Taiwan, since 2012. Her current research interests include multimedia content analysis, action recognition, and healthcare applications.

Wen-Huang Cheng (M'14) received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2002 and 2004, respectively, where he received the Ph.D. (Hons.) degree from the Graduate Institute of Networking and Multimedia, in 2008.

He is an Assistant Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, where he is the Founding Leader with the Multimedia Computing Laboratory and an Adjunct Assistant Research Fellow with the Institute of Information Science. Before joining Academia Sinica, he was a Principal Researcher with MagicLabs, HTC Corporation, Taoyuan, Taiwan, from 2009 to 2010. His current research interests include multimedia content analysis, computer vision, mobile multimedia computing, and human computer interaction.

Dr. Cheng has received numerous research awards, including the Outstanding Young Scholar Awards from the Ministry of Science and Technology in 2014 and 2012, the Outstanding Social Youth of Taipei Municipal in 2014, the Best Reviewer Award from the 2013 Pacific-Rim Conference on Multimedia, and the Best Poster Paper Award from the 2012 International Conference on 3-D Systems and Applications. He supervised the post-doctoral fellows to award the Academia Sinica Post-Doctoral Fellowship in 2013 and 2011.

Che-Han Chang received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree from the Graduate Institute of Networking and Multimedia.

His current research interests include image processing and computer vision.

Jui-Hsin Lai received the B.S. degree in electronics engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 2005, and the Ph.D. degree from the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan, in 2011.

From 2007 to 2011, he was a Project Leader with Yotta-Labs, an IC design house in Taipei, where he was involved in designing the algorithm and hardware architecture of vision-based object tracking, face detection, and recognition. Since 2011, he has been a Post-Doctoral Fellow with the Graduate Institute of Networking and Multimedia, NTU. His current research interests include applications of interactive multimedia, computer vision, sports video, video/image processing, and VLSI architecture design of multimedia processing.

Ja-Ling Wu (F'08) received the B.S. degree in EE from TamKang University, Taipei, Taiwan, in 1979, and the M.S. and Ph.D. degree in EE from the Tatung Institute of Technology, Taipei, Taiwan, in 1981 and 1986, respectively.

He has been a Professor with the Department of Computer Science and Information Engineering, National Taiwan University (NTU), Taipei, Taiwan, since 1996. From 2004 to 2007, he was appointed to be the first Head of the Graduate Institute of Networking and Multimedia, NTU. He was selected to be one of the Lifetime Distinguished Professors of NTU, in 2006.