

User Performance Evaluation and Real-Time Guidance in Cloud-Based Physical Therapy Monitoring and Guidance System

Wenchuan Wei, Yao Lu, Eric Rhoden, and Sujit Dey

Abstract—The effectiveness of traditional physical therapy may be limited by the sparsity of time a patient can spend with the physical therapist (PT) and the inherent difficulty of self-training given the paper/figure/video instructions provided to the patient with no way to monitor and ensure compliance with the instructions. In this paper, we propose a cloud-based physical therapy monitoring and guidance system. It is able to record the actions of the PT as he/she demonstrates a task to the patient in an offline session, and render the PT as an avatar. The patient can later train himself by following the PT avatar and getting real-time guidance on his/her device. Since the PT and user (patient) motion sequences may be misaligned due to human reaction and network delays, we propose a Gesture-Based Dynamic Time Warping algorithm that can segment the user motion sequence into gestures, and align and evaluate the gesture sub-sequences, all in real time. We develop an evaluation model to quantify user performance based on different criteria provided by the PT for a task, trained with offline subjective test data consisting of user performance and physical therapist scores. Moreover, we design three types of guidance which can be provided after each gesture based on user score, and conduct subjective tests to validate their effectiveness. Experiments with multiple subjects show that the proposed system can effectively train patients, give accurate evaluation scores, and provide real-time guidance which helps the patients learn the tasks and reach the satisfactory score with less time.

Index Terms— dynamic time warping, gesture segmentation, motion data alignment, physical therapy, real-time guidance

I. INTRODUCTION

In recent years, the emergence of various medical sensors and monitoring devices has led to the widespread development of smart healthcare which can provide cheaper, faster, and more effective monitoring and treatment for patients [1]-[5]. As a widely used type of rehabilitation in the treatment of many diseases, physical therapy is a promising field in smart healthcare applications. Traditional physical therapy involving training in professional therapy sessions can be expensive and even unaffordable for many patients. Even if patients are instructed in therapy sessions, they need to practice at home by following paper or figure instructions, which cannot provide effective feedback and track patient performance. To address this problem, virtual training systems based on rendering technologies and motion capture sensors such as Microsoft Kinect [6] are being developed [7], [8]. In the

meantime, the use of mobile devices has become pervasive – for example, in June 2016, mobile applications and browsers accounted for 67% of digital media time spent in the United States [9]. In addition, cloud computing has started being used as an alternative approach for mobile health applications [10], computer games [11], etc., to make up the inherent hardware constraint of mobile devices in memory, graphics processing and power supply when running heavy multimedia and security algorithms. In cloud-based mobile applications, all the data and videos are processed and rendered on the cloud, which makes it superior to local processing on desktop computers for its portability across multiple platforms. Thus, this solution can enable users to use the system at home or away, e.g. at hotels while traveling, making it more flexible and usable. In this paper, we combine 1) rendering technology, 2) motion capture based on Microsoft Kinect and 3) cloud computing for mobile devices to propose a cloud-based real-time physical therapy instruction, monitoring and guidance system. The proposed system enables a user to be trained by following a pre-recorded avatar instructor, monitors and quantifiably measures user performance, and provides real-time textual and visual guidance on his/her mobile device as needed to improve the user’s performance. Note that in this paper, we use the terms “user” and “patient” interchangeably.

The architecture of the proposed cloud-based physical therapy monitoring and guidance system is shown in Fig. 1. Note that the physical therapy tasks discussed in this paper are movement based tasks. Fig. 1(a) shows the offline session, in which a physical therapist (PT) defines the criteria and satisfactory score for a task, and also demonstrates the task, with his/her motion data captured by the Kinect sensor and his/her avatar recorded and trained on a game development platform Unity [12]. (To avoid confusion, we use the abbreviation “PT” to refer to the PT avatar showing on the user device, and use “physical therapist” to refer to the real physical therapist colleague in this project team.) For each task, an evaluation model is trained from a subjective test, which is used to evaluate the user’s performance on this task. Fig. 1(b) shows the online home session. A training video is transmitted through a wireless network to the user device. The user watches the training video and tries to follow the task. Simultaneously, his/her movements are captured by Kinect

and uploaded to the cloud. On the cloud, the proposed Gesture-Based Dynamic Time Warping algorithm segments the user’s motion sequence into gestures and aligns the motion data of the PT and user in real time. User’s accuracy is determined by transforming the user’s errors into an overall score using the evaluation model obtained from the offline session. The alignment results are processed by a guidance logic. The user can progress to the next task if and when his/her accuracy reaches a satisfactory score, otherwise a guidance video is rendered and transmitted to the user device to help the user calibrate his/her movements.

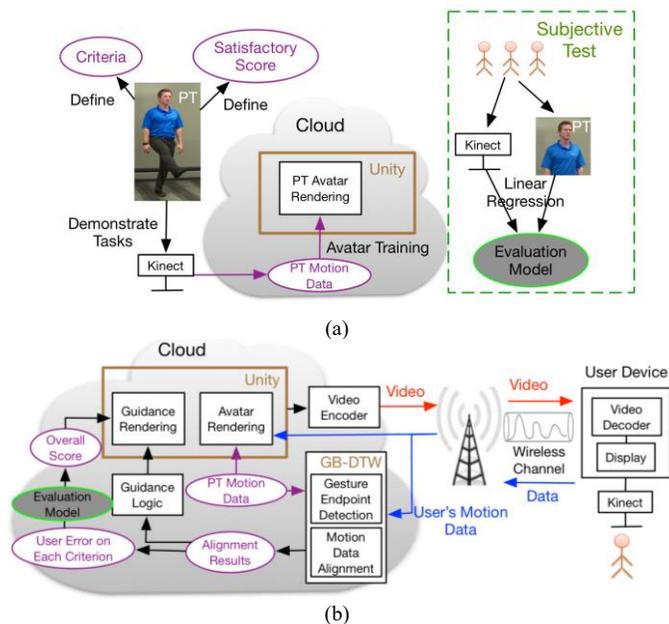


Fig. 1. Architecture of cloud-based physical therapy monitoring and guidance system. (a) Offline session. (b) User home session.

The proposed system has the ability to more effectively and efficiently train people for different types of tasks, like knee rehabilitation, shoulder stretches, etc. Although other avatar-based training systems exist, our system provides real-time guidance rather than just providing scores. This feature allows the system to cater to the abilities of the user and to react to the user’s performance by demonstrating the necessary adjustments to establish optimal conditions. In essence, our system is dynamic, allowing every user experience to be distinct. Moreover, together with the offline step of capturing and training an avatar for the PT tasks customized to a particular user, the proposed system enables personalized physical therapy training.

Although the platform has the advantages as mentioned above, human reaction delay (delay by user to follow instructions) and wireless network delay (which may delay when the cloud rendered avatar video reaches the user device) may cause challenges for correctly calculating the accuracy of the user’s movements compared to the PT’s movements. In particular, the delay may cause the two motion sequences to be misaligned with each other and make it difficult to judge whether the user is following the PT correctly. Therefore, we apply Dynamic Time Warping (DTW) algorithm to address the

problem of motion data misalignment. Considering the fact that DTW can only be applied after the user finishes the whole task, we further propose the Gesture-Based Dynamic Time Warping algorithm to segment the whole user motion sequence into gestures to enable real-time evaluation and guidance for the user. To evaluate the user’s performance correctly, an evaluation model is trained by collecting data from subjective test and based on the professional advice of the physical therapist in our team. To help the user improve accuracy, we design visual/textual/combined guidance and conduct subjective test to validate their effectiveness. We have implemented the proposed algorithms in a prototype avatar based real-time guidance system and conducted experiments using wireless network profiles and on a real cloud environment. Experimental results show the performance advantage of our proposed method over other alignment methods, as well as the feasibility and effectiveness of our proposed cloud-based physical therapy monitoring and guidance system.

A preliminary version of this work has been reported in [13]. Compared with [13], we have developed a new real-time monitoring and guidance system in this paper using Unity [12], which enables more effective avatar modeling, user performance tracking, and guidance design and delivery. The motion data are extended from one dimension to multi dimensions. In user performance evaluation, we present a new Gesture-Based Dynamic Time Warping algorithm which significantly enhances the accuracy of gesture segmentation and reduces segmentation delay, compared to the algorithm we presented in [13]. (In the rest of this paper, we use GB-DTW0 to refer to the algorithm proposed in [13] and GB-DTW-A to refer to the new algorithm proposed in this paper where “A” means more accurate segmentation.) Experimental results are provided to demonstrate the superior performance of the new GB-DTW-A algorithm. Furthermore, the user performance evaluation model is completely redesigned based on a procedure involving subjective testing. A new guidance system is designed which can provide more intuitive and detailed guidance. Effectiveness of the proposed real-time guidance, not discussed in [13], is validated with a new subjective study. The main overlap of this paper with [13] is in the introduction of the classical DTW algorithm (Section IV-A) and part of the experimental results in Section V-A.

The rest of the paper is organized as follows: Section II reviews related work about automatic training systems for physical therapy and their related user performance evaluation techniques and guidance system. In Section III, we introduce the construction of motion data and the data misalignment problem. Section IV proposes the data alignment approach and the evaluation model for the user’s performance, as well as the guidance design in the proposed system. Section V presents the experimental results of motion data alignment and performance evaluation using real network profiles and on a real cloud environment, and also validates the effectiveness of guidance. Section VI concludes the paper and discusses future work.

II. RELATED WORK

A. Automatic Training System for Physical Therapy

Physical therapy is a widely used type of rehabilitation in the treatment of many diseases. Normally, patients are instructed by specialists in physical therapy sessions and then expected to practice the activities at home, in most cases following paper/figure instructions they are given in the sessions. However, they cannot get useful feedback about their performance and have no idea how to improve their training without the supervision of the professional physical therapists. To address this problem, some automatic training systems have been created to train people at home. In [8], the authors use the marker-based optical motion capture system Vicon and prove its effectiveness in gait analysis on subjects with hemiparesis caused by stroke. A wearable electronic device called Pt Viz is developed for knee rehabilitation [14]. Furthermore, Microsoft Kinect sensor is proved of high accuracy and more convenient in detecting the human skeleton compared with wearable devices [15]. Authors in [16] develop a game-based rehabilitation system using Kinect for balance training. In [17], Kinect is used to track arm movements to help young adults with motor disabilities. In our proposed system, Kinect is used to track physical therapy tasks for its efficiency in full-body and limb tracking, as well as being readily available, easy to setup, and low-cost. Besides, our proposed system is superior to the above Kinect-based systems for its high accuracy and reliability in user performance evaluation and guidance design. In [16] and [17], Kinect is used primarily to motivate the users, without accurate feedback on the user's performance. Our proposed evaluation method addresses two kinds of delay problem in the user motion sequence, which will be discussed in the following sections.

B. User Performance Evaluation

In physical therapy, patients' movements need to be carefully controlled due to their reduced mobility and the potential for re-injury. Therefore, user performance evaluation is an important part in these automatic training systems to remind patients of any incorrect motion. To evaluate the user's performance, authors in [18] propose to compare the skeletons of the trainee and the trainer tracked by Kinect sensor. First, skeleton of the trainee is scaled by resizing each bone to match the size of the corresponding bone of the trainer. Then the two skeletons are aligned by aligning the hips which are considered to be the hierarchical center of the skeleton. Finally, the trainee's performance can be evaluated by calculating the Euclidean distance between the trainee's and trainer's joints.

However, the assumption of this approach is that the trainee follows the trainer timely since they use a window of 0.5s for any target frame to search for the best matching posture. For some challenging tasks, it might be difficulty for the user, especially for patients with injuries, to catch up with the trainer's movements. In this case, motion data of the trainer

and the trainee are mismatched and the best matching posture cannot be found within the 0.5s window.

To address the misalignment problem, authors in [19] propose to use Maximum Cross Correlation (MCC) to calculate the time shift between the standard/expected motion sequence and the user's motion sequence. Then by shifting the user's motion sequence by the estimated time shift, the two sequences are aligned and their similarity can be calculated. However, this approach assumes uniform delay during the user's movements and cannot address the problem of motion data distortion, which will be discussed in Section III-B.

In [20], a training system based on wearable sensor use DTW to detect and identify correct and incorrect executions in an exercise. It is aimed at finding the best match of the user's execution among some correct and incorrect templates to judge the user's performance and give the error type if any. However, error templates can hardly cover all the mistakes patients may make, and computation increases with more templates. Besides, it can only be applied offline when the entire user motion sequence is obtained. In comparison, the proposed system does not need any pre-recorded error template. Besides, the proposed GB-DTW-A algorithm enables real-time evaluation and guidance for the user.

C. Guidance Design

To help the user improve performance, many types of guidance system have been designed. OctoPocus [21] and ShadowGuides [22] teach user gestures and movements on touch screens. LightGuide [23] projects guidance hints directly on a user's body to guide the user in completing the desired motion. In [14], wearable sensor made of lighted fabric visualizes the correct knee angle for knee rehabilitation exercises. BASE [24] based on kinematic sensor designed for older adults displays colored markers overlaid on the body to show the user's position and target position. In [18], an augmented reality mirror and colored circles/lines overlaid on the user's body are used to instruct the user and label incorrect movements. In [25], an on-screen "Wedge" visualization overlaid on top of the user's body shows the plane and range of movement, joint positions and angles, and extent of movement.

Most of the above guidance systems instruct the user on how to perform the task correctly by specifying the target body position and telling the user whether he/she has reached the target or not. However, we would like to develop a guidance system that is more adaptive and personalized for each task and also for each user. In the proposed system, guidance is provided based on criteria specially designed for each task by the physical therapist, instead of simply comparing the complete skeletons of the PT and user and showing the mismatched joints. Moreover, the proposed system can also decide whether the user needs to be guided according to the user's performance and a satisfactory score set by the physical therapist, which avoids overwhelming instructions in training.

III. MOTION DATA CONSTRUCTION AND DATA MISALIGNMENT PROBLEM

In the proposed system, Kinect captures 25 joints with 3-D coordinates for each joint [26]. However, only some parts of these joints are deemed important for a specific task. In this section, we will introduce how to construct the motion data for a task and the motion data misalignment problem in the system.

A. Motion Data Construction

For a given task, the physical therapist defines several criteria and the tolerable error threshold for each criterion, which need to be translated into motion features. Motion features are quantities that are derived from the joint coordinates captured by Kinect, such as joint positions, joint angles, joint velocity, etc. For example, in the shoulder abduction task, arm height or shoulder angle (i.e., angle between the arm and the vertical direction) can be a motion feature which indicates whether the user raises the arm highly enough. Considering the difference in body size, we use normalized features, like angles, to build the motion data. The first three columns in Table I show the examples of some criteria defined by the physical therapist, the corresponding motion features, and the tolerable error threshold for a leg lift task.

TABLE I

EXAMPLES OF TASK CRITERIA AND MOTION FEATURES OF LEG LIFT TASK

Criterion	Motion Features	Error Threshold	Feature Type
“Lift right leg to the required height”	Angle between right leg and vertical direction: 60^0	$\pm 5^0$	Time-varying
“Keep right knee straight”	Angle between right thigh and right shank: 180^0	$\pm 10^0$	Constraint
“Keep right leg in front of the body”	Angle between right leg and the patient’s right direction: 90^0	$\pm 10^0$	Constraint

Moreover, there are two types of features: time-varying features and constraint features. In a task, the patient is instructed to move some parts of his/her body, and keep some other parts stationary in the meantime. Time-varying features are features which represent the body’s movements in this task. Constraint features represent the other body parts which should be kept stationary during the task. The fourth column in Table I shows the corresponding feature type of each criterion in the leg lift task.

For a given task, the physical therapist defines N_v time-varying features and N_c constraint features. Time-varying motion data F^v for this task can be obtained by combining all the time-varying motion features of each frame.

$$F^v = \begin{bmatrix} f_{1,1}^v & f_{1,2}^v & \cdots & f_{1,N_v}^v \\ f_{2,1}^v & f_{2,2}^v & \cdots & f_{2,N_v}^v \\ \vdots & \vdots & \ddots & \vdots \\ f_{T,1}^v & f_{T,2}^v & \cdots & f_{T,N_v}^v \end{bmatrix}, \quad (1)$$

where T is the number of frames, $f_{t,i}^v$ is the i -th time-varying feature in frame t . Similarly, constraint motion data F^c is

$$F^c = \begin{bmatrix} f_{1,1}^c & f_{1,2}^c & \cdots & f_{1,N_c}^c \\ f_{2,1}^c & f_{2,2}^c & \cdots & f_{2,N_c}^c \\ \vdots & \vdots & \ddots & \vdots \\ f_{T,1}^c & f_{T,2}^c & \cdots & f_{T,N_c}^c \end{bmatrix}, \quad (2)$$

where $f_{t,j}^c$ is the j -th constraint feature in frame t .

B. Motion Data Misalignment Problem

Given the motion data of the PT and the user, we calculate the similarity of the two sequences to evaluate the performance of the user. However, comparing the two sequences directly is unreliable due to the potential data misalignment caused by delay. There are mainly two kinds of delay in the system: 1) human reaction delay, which means that it may take the user some time to react to the demonstration task before following it, 2) network delay, which results from the wireless network when transmitting the training video from the cloud to the user device.

Human reaction delay and network delay cause two types of motion data misalignment problem: time shift and data distortion. In the rest of this section, we will discuss these two types of data misalignment problem, and discuss the problems the existing technique MCC [19] has in addressing the misalignment between the two sequences.

1) Time Shift Delay

When human reaction delay and network delay are uniform in a training task, there is only time shift between the PT’s and the user’s motion data. In this case MCC can be used to estimate the time shift and align the two sequences. For two discrete-time signals f and g , their cross correlation $R_{f,g}(n)$ is defined by

$$R_{f,g}(n) = \sum_{m=-\infty}^{\infty} f^*(m)g(m+n), \quad (3)$$

and the time shift τ of the two sequences is estimated as the position of maximum cross correlation

$$\tau = \arg \max_n \{R_{f,g}(n)\}. \quad (4)$$

For those tasks including multiple separate gestures, the time shift might be different for these gestures and need to be calculated separately. Here we define a gesture as a subsequence that represents an independent subtask, e.g., one-time shoulder abduction and adduction. Gestures in a training task are segmented manually by the physical therapist. Fig. 2 shows a simple example of the PT and user’s motion data in a task of three gestures. For each gesture, the user follows the

PT avatar to perform shoulder abduction and adduction. Fig. 2(b) shows the angle between the left arm and the vertical direction as an example of the motion feature. Suppose that the user performs each gesture with delay τ_1 , τ_2 and τ_3 ($\tau_1 \neq \tau_2 \neq \tau_3$), they can be estimated using MCC and the two sequences can be aligned by shifting each gesture by the corresponding estimated delay.

2) Motion Data Distortion

In many cases, human reaction delay and network delay may not be uniform. The user may not be able to follow the task timely or perform some incorrect motion when the task is difficult for him/her. For example, when following a task of 2 seconds, it takes a user 1s to react to the instructions and another 1s to complete the task since he realizes that he is behind. In this case the user's reaction delay is not uniform ($\text{delay} = 1\text{s}$ when $t \leq 1\text{s}$, $\text{delay} < 1\text{s}$ when $1\text{s} < t < 2\text{s}$, and $\text{delay} = 0$ when $t = 2\text{s}$). Besides, the user's valid motion sequence (1s) is shorter than the PT's (2s), so shifting one sequence by the estimated delay cannot effectively align them. Network delay may also be not uniform due to many factors, such as varying bandwidth and network load. Although some response time management techniques have been developed [27], the network delay in cloud mobile applications cannot be eliminated. Therefore, under the influence of fluctuating network delay or when the user is following some difficult tasks, the user's motion data might be distorted compared with the PT's. Fig. 3 shows the motion data of the same task as Fig. 2, but with both time shift delay and motion data distortion. In this case, using MCC to shift the user's sequence by an estimated delay is unreliable. To calculate the similarity between the two sequences effectively, we need to find an optimal way to align them.

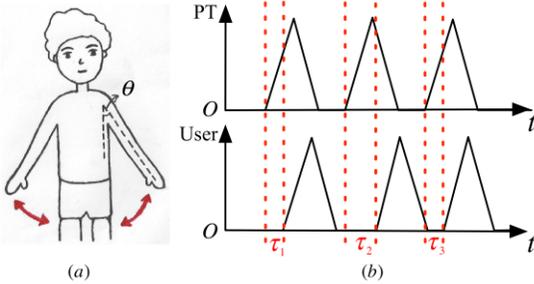


Fig. 2. (a) Shoulder abduction and adduction. (b) Motion data (i.e., angle between left arm and the vertical direction) of the PT and user for three gestures with only time shift delay. Delay for each gesture is τ_1 , τ_2 , τ_3 .

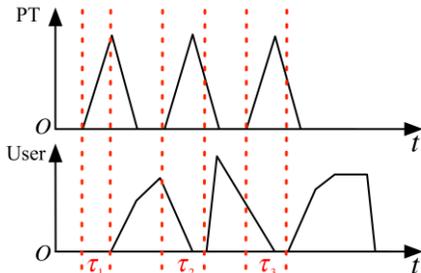


Fig. 3. Motion data (i.e., angle between left arm and vertical direction) of the PT and the user with both time shift delay and motion data distortion.

IV. MOTION DATA ALIGNMENT AND USER PERFORMANCE EVALUATION

To solve the data misalignment problem and evaluate the user's performance correctly, we propose a DTW-based data alignment and evaluation method. Section IV-A introduces the principle of classical DTW and its use in the proposed system. Section IV-B proposes the GB-DTW-A algorithm which segments user gestures so that data alignment can be done in real time based on each gesture, and introduces the enhancements of GB-DTW-A compared with the original GB-DTW0 algorithm [13]. In Section IV-C, we discuss how to evaluate the user's performance according to the alignment results of GB-DTW-A. Finally, Section IV-D introduces visual and textual guidance in the proposed system and discusses how to provide effective guidance for the user.

A. Dynamic Time Warping

DTW is a dynamic programming algorithm that is widely used in speech processing [28]. It measures the similarity between two sequences $A = \{a_1, a_2, \dots, a_m\}$ and $B = \{b_1, b_2, \dots, b_n\}$ by calculating their minimum distance. To calculate the minimum distance, an $m \times n$ distance matrix D is defined where $D(i, j)$ is the Euclidean distance between a_i and b_j .

$$D(i, j) = \|a_i - b_j\|. \quad (5)$$

To find the best alignment between A and B , a continuous warping path through the distance matrix D should be found such that the sum of items on the path is minimized. Hence, this optimal path stands for the optimal mapping between A and B such that their distance is minimized. This path is defined as $P = \{p_1, p_2, \dots, p_q\}$ where $\max\{m, n\} \leq q \leq m + n - 1$ and $p_k = (x_k, y_k)$ indicates that a_{x_k} is aligned with b_{y_k} . Moreover, this path is subject to the following constraints.

- Boundary constraint: $p_1 = (1, 1)$ and $p_q = (m, n)$.
- Monotonic constraint: $x_{k+1} \geq x_k$ and $y_{k+1} \geq y_k$.
- Continuity constraint: $x_{k+1} - x_k \leq 1$ and $y_{k+1} - y_k \leq 1$.

To find the optimal path, an $m \times n$ accumulative distance matrix S is constructed where $S(i, j)$ is the minimum accumulative distance from $(1, 1)$ to (i, j) . The accumulative distance matrix S can be represented as

$$S(i, j) = D(i, j) + \min \begin{cases} S(i-1, j-1) \\ S(i, j-1) \\ S(i-1, j) \end{cases}, \quad (6)$$

and $S(m, n)$ is defined as the DTW distance between A and B [29]; smaller DTW distance indicates that the two sequences are more similar. The optimal warping path can be found by backtracking from (m, n) to $(1, 1)$ and this path indicates the best way to align the two sequences. Time complexity of the DTW method is $O(mn)$. Fig. 4(a) shows an example of two sequences A and B . The purple marked elements construct a path from $(1, 1)$ to (m, n) on which the accumulative distance

is minimized. It is the optimal warping path between A and B . Fig. 4(b) shows the corresponding alignment given by the optimal path in Fig. 4(a). For example, a_1 is aligned with b_1 , a_2 and a_3 are aligned with b_2 .

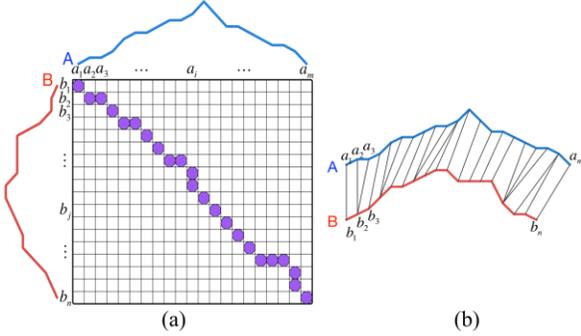


Fig. 4. (a) Warping path of DTW on sequence A and B . (b) Alignment results between A and B .

In the proposed system, DTW can be used to find out the optimal alignment between the PT’s and user’s movements. As mentioned in Section III-A, there are two types of motion data: time-varying motion data F^v and constraint motion data F^c . For time-varying motion data F^v , delay problems mentioned in Section III-B may cause the data to be misaligned with each other. Therefore, DTW can be applied to the PT’s and user’s time-varying motion data to find out an optimal warping path $P = \{p_1, p_2, \dots, p_q\}$, where $p_k = (x_k, y_k)$ indicates that the user’s performance in frame y_k matches PT’s movement in frame x_k . Constraint motion data vs. time are horizontal lines (e.g., the knee angle vs. time is a horizontal line at 180 degrees for criterion “keep knee straight” in Table I and DTW cannot be used to align them. Constraint motion data are aligned using the DTW alignment results of time-varying motion data. Consequently, based on the alignment results, the user’s performance can be evaluated by comparing his/her movements with the PT’s demonstration movements.

B. Real-Time Gesture Segmentation Based on DTW

Although DTW is an effective way to find out the optimal alignment between the PT and user’s motion sequences, it works only after the two motion sequences are obtained, that is, after the user finishes the entire task. In the proposed system, we would like to provide real-time evaluation for the user after he/she finishes each gesture, thus real-time gesture segmentation is needed during the user’s performance. There has been numerous research in the field of gesture segmentation, including methods based on machine learning, signal processing [30], [31], etc. In this work, since DTW can be used to align the motion sequences, we further propose a variant of DTW called GB-DTW-A so that gesture segmentation can be implemented in the process of DTW. We next present the details of GB-DTW-A.

For a given task, gestures in the PT’s motion sequence have been pre-defined and segmented, which will be used as the ground truth to segment user’s gestures. Suppose that $A_1 = \{a_1, a_2, \dots, a_{m_1}\}$ is defined as the first gesture in the PT’s motion

sequence A . Then we would like to use DTW to find a subsequence $B_1 = \{b_1, b_2, \dots, b_k\}$ ($2 \leq k \leq n$) of the user’s motion data B that matches the PT’s gesture A_1 best. Since the DTW distance $S(m_1, k)$ represents the similarity between A_1 and B_1 , the optimal endpoint n_1 of the user’s gesture should be the position with the minimum DTW distance.

$$n_1 = \arg \min_{2 \leq k \leq n} \{S(m_1, k)\}. \quad (7)$$

In [29], the Subsequence DTW algorithm searches the entire user sequence B to find out the global optimum n_1 . However, it works only after the user completes the entire task, which means that it is not real-time. Here we propose a new approach to estimate the global optimum by testing each local optimum. Firstly, we define a normalized distance function $T(k) = S(m_1, k) / (\sum_{i=1}^{m_1} a_i)$, where $\sum_{i=1}^{m_1} a_i$ is the sum of PT’s motion data on this gesture. Then $T(k)$ can be used as a uniform similarity metric for different gestures. For a local optimum k^* , we propose the following conditions to check whether it is the global optimum.

Condition 1: k^* is the current global optimum, i.e., $T(k^*) \leq T(k)$ for any $k < k^*$.

Condition 2: The normalized distance between A_1 and B_1 is below a threshold, i.e., $T(k^*) < \tau$.

Condition 1 is a necessary condition for the global optimum. If Condition 1 is not satisfied, we continue to search and check the next local optimum. In Condition 2, if the threshold is set strict (i.e., τ is low), it fails to consider the possibility of user’s poor performance even if the user has completed the gesture. If the threshold is set loose (i.e., τ is high), $T(k^*) < \tau$ may be satisfied at some local optimums before the user completes the gesture. To solve this problem, we propose a dual-threshold strategy as follows. In Condition 2, a strict threshold τ_S is used. Therefore, Condition 2 is used to detect the global optimum when the user is following the PT accurately. If a local optimum satisfies both Condition 1 and Condition 2, it can be estimated as the global optimum. If only Condition 1 is satisfied and Condition 2 is not satisfied, we further use the following method to check whether k^* may be the endpoint of the user’s gesture. If k^* is the global optimum n_1 , B_1 is the best match with A_1 . When the user completes one gesture, he/she may stay in the ending posture for several frames, so the following frames $\{b_{n_1+1}, b_{n_1+2}, \dots\}$ will be quite close to b_{n_1} . Based on the above observation, we propose the following empirical evidence. For the global optimum n_1 , all of its following r frames $\{b_{n_1+1}, b_{n_1+2}, \dots, b_{n_1+r}\}$ tend to be aligned with a_{m_1} in DTW. In other words, for frame $n_1 + j$ ($j = 1, 2, \dots, r$), (6) becomes

$$S'(m_1, n_1 + j) = D(m_1, n_1 + j) + S'(m_1, n_1 + j - 1), \quad (8)$$

For the r frames following a local optimum k^* , we calculate the DTW distances $S_{true} = \{S(m_1, k^*+1), S(m_1, k^*+2), \dots, S(m_1, k^*+r)\}$. In the meantime, we compute $S_{assumption} = \{S'(m_1, k^*+1),$

$S'(m_1, k^*+2), \dots, S'(m_1, k^*+r)\}$ using (8). The relative error between S_{true} and $S_{assumption}$ is

$$error = \left| \frac{S_{assumption} - S_{true}}{S_{true}} \right|. \quad (9)$$

Then we propose Condition 3 to further test a local optimum k^* in case Condition 2 is not satisfied.

Condition 3: The relative error between S_{true} and $S_{assumption}$ is below an error tolerance threshold δ , i.e., $\text{Mean}(error) < \delta$. Besides, the normalized distance between A_1 and B_1 is below a loose threshold τ_L , i.e., $T(k^*) < \tau_L$.

Condition 3 is used to detect the global optimum for the user's poor performance. When the user performs the task, the normalized distance $T(k)$ is calculated for each frame k . For any local optimum k^* , it is estimated as the global optimum if it satisfies Condition 1 and 2. If Condition 2 is not satisfied, Condition 3 is further used to test it.

However, it is still possible that a true global optimum n_1 does not meet Condition 2 or 3. If we continue searching the following frames after n_1 , $T(k)$ will keep increasing and we cannot obtain the correct segmentation result even until the end of the task. To stop the searching timely, we propose Condition 4 to decide whether the current frame k is behind the global optimum n_1 .

Condition 4: $T(k) > T(1)$ and there exists $k_0 < k$ such that $T(k_0) < \tau_M$.

In Condition 4, $T(k_0) < \tau_M$ is used to exclude the situation where $T(k)$ may be increasing for the first several frames. If frame k satisfies Condition 4, the search should be stopped and the current global optimum (i.e., the minimum point among $T(1) \sim T(k)$) can be estimated as the global optimum. The pseudo-code for the proposed GB-DTW-A algorithm is shown in Fig. 5.

Compared with GB-DTW0 proposed in [13], the new GB-DTW-A algorithm achieves higher segmentation accuracy and less segmentation delay. In GB-DTW0, only Condition 3 is used to test local optimums. However, the single threshold δ is sensitive to the user's performance. Fig. 6 shows an example where the task and motion feature are the same as Fig. 2. Fig. 6(a) shows the motion sequence of a PT's gesture, and Fig. 6(b)(c) show the motion data of two users, where E_1 and E_2 are the endpoints of their gestures. User 1 follows the PT accurately, so the DTW distance between PT and user 1 is small. For the true gesture endpoint E_1 , the relative error in (9) may be high since S_{true} is small. In this case, the threshold δ should be higher to allow E_1 to be detected as the global optimum. User 2 is performing poorly (not following PT accurately), so the DTW distance is large. Point A is a local optimum of the DTW distance, but not the gesture endpoint. For point A , the relative error in (9) may be small since S_{true} is large. To avoid mistakenly detecting A as the global optimum, δ should be set lower. Therefore, a uniform threshold δ for all users may result in segmentation errors. In contrast, the dual-threshold strategy proposed in GB-DTW-A can be used for all types of user performance, and therefore reduce the segmentation errors. Besides, the segmentation delay (i.e., the

delay between the true gesture endpoint and the time when the segmentation is completed) of GB-DTW0 is at least r frames since Condition 3 needs to check r frames following the gesture endpoint. In GB-DTW-A, Condition 1 and 2 can be checked in real time without any delay. Condition 3 is checked only if Condition 2 is not satisfied. Moreover, Condition 4 provides a way to stop the searching in time when we miss the global optimum instead of searching to the end of the task (which is used by GB-DTW0). Thus GB-DTW-A also reduces the segmentation delay compared with GB-DTW0. Details about the comparison results between these two algorithms are provided in Section V-B.

Algorithm Gesture-Based Dynamic Time Warping (GB-DTW-A)

Input: PT's gesture A_1 , user's motion sequence $B = \{b_1, b_2, \dots, b_n\}$

Output: Endpoint of user's gesture

Initialization: $curMin = Inf, curMinIndex = -1, flag = false$

1. **for** each frame k in sequence B
 2. **if** k is a local minimum and $T(k) < curMin$
 3. **if** $T(k) < \tau_S$
 4. **return** k
 5. **else**
 6. calculate S_{true} and $S_{assumption}$
 7. $error = \left| \frac{S_{assumption} - S_{true}}{S_{true}} \right|$
 8. **if** $\text{Mean}(error) < \delta$ and $T(k) < \tau_L$
 9. **return** k
 10. **end if**
 11. **end if**
 12. **end if**
 13. **if** $T(k) > T(1)$ and $curMinIndex > 0$ and $flag == true$
 14. **return** $curMinIndex$
 15. **end if**
 16. **if** $T(k) < curMin$
 17. $curMin = T(k)$ and $curMinIndex = k$
 18. **end if**
 19. **if** $flag == false$ and $T(k) < \tau_M$
 20. $flag = true$
 21. **end if**
 22. **end for**
 23. **return** $curMinIndex$
-

Fig. 5. Pseudo-code of GB-DTW-A algorithm.

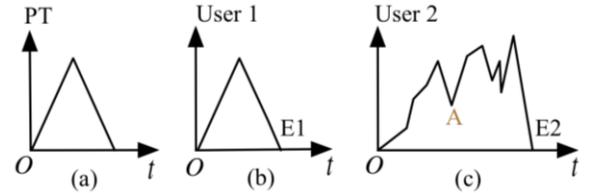


Fig. 6. (a) PT's motion sequence. (b) User 1's motion sequence with accurate performance. (c) User 2's motion sequence with poor performance. A is a local optimum of the DTW distance.

Using the above approach, gesture segmentation is implemented in the process of DTW. If $B_1 = \{b_1, b_2, \dots, b_{n_1}\}$ is determined as the user's gesture related to the PT's gesture A_1 , DTW can be conducted from the new starting point $(m_1 + 1, n_1 + 1)$. Fig. 7 shows the example of applying GB-DTW-A on the same sequences as Fig. 4. Suppose that there are four gestures in the task, segmentation allows DTW to be performed

separately for each gesture. The shaded area is indicative of the computation cost for each gesture.

For each gesture, Condition 1 and 2 can be checked on each local optimum in constant time. For a task with g gestures, each PT's gesture contains m/g frames and each user's gesture contains n/g frames on average. The complexity of GB-DTW-A on each gesture is $O(mn/g^2)$. For Condition 3, r more frames following the local optimum need to be tested. The extra complexity to test local optimum is $O(mr/g)$. So the average complexity of GB-DTW-A is

$$O\left(g \times \left(\frac{mn}{g^2} + \frac{mr}{g}\right)\right) = O\left(m\left(\frac{n}{g} + r\right)\right) = O\left(\frac{mn}{g}\right) \ll O(mn). \quad (10)$$

When the number of gestures g in the sequence is large, the proposed GB-DTW-A algorithm can significantly decrease the computation complexity compared to classical DTW on the entire sequence. If real-time detection fails, which means that the true global optimum does not meet Condition 2 or 3, Condition 4 is used to break the search and output the correct segmentation result, although with some delay. In this case, the computation complexity increases. If the segmentation is delayed to the end of the entire task in the worst case, the complexity becomes $O(mn)$. However, it is shown in Section V-B that this worst situation happens very rarely. In most cases, the segmentation delay is low and the complexity is close to $O(mn/g)$.

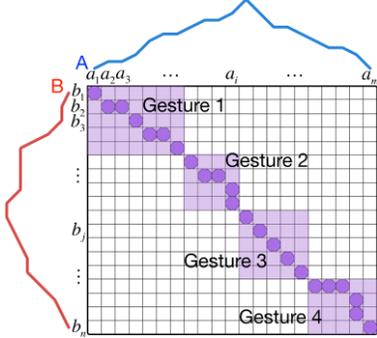


Fig. 7. Average computation complexity of GB-DTW-A in a task of four gestures.

C. GB-DTW-A Based User Performance Evaluation

In this section, we will discuss how GB-DTW-A can be applied to evaluate the user's performance. As discussed in the last section, GB-DTW-A aligns motion sequences as soon as the user completes a gesture, instead of waiting until the entire task is over, with much less complexity compared with classical DTW. Then based on the alignment results, we can check the user's error on each criterion by comparing his/her motion data with the matched PT's motion data, and calculate an overall evaluation score for his/her performance on the previous gesture.

1) GB-DTW-A Based User Error for Each Criterion

For each criterion in a task (see examples in Table I), we denote $A = \{a_1, a_2, \dots, a_m\}$ as the PT's motion data and $B = \{b_1,$

$b_2, \dots, b_n\}$ as the user's motion data. An optimal path $P = \{p_1, p_2, \dots, p_q\}$ which indicates the optimal alignment between A and B has been calculated by applying GB-DTW on the time-varying motion data.

To measure the user's error, first we need to discuss different alignment types in P . We define the monotonicity of a subsequence $A^* = \{a_i, a_{i+1}, \dots, a_{i+w-1}\}$ as follows. If all the elements in A^* are monotonic (i.e. keep increasing or decreasing) then A^* is monotonic, otherwise it is non-monotonic. When multiple PT frames $A^* = \{a_i, a_{i+1}, \dots, a_{i+w-1}\}$ are aligned with one single user frame b_j , there are two different cases. (a) If A^* is monotonic, it means that the effects of multiple frames in A^* are similar to the effect of b_j , which indicates that the user moved faster than the PT at that time. (b) If A^* is non-monotonic, it means that some back and forth PT movements are simplified as one single frame b_j in the user's performance, thus the user's movement is incomplete for this back and forth motion. Similarly, if one single PT frame is aligned with multiple user frames, we can judge whether the user is slower or overdoes the movement. (Note that the cause for the user to be slow might also be due to receiving the training video delayed due to the wireless network, that is, effect of network delay.) Table II and Fig. 8 illustrates the five alignment types in DTW. For example, in type 1 the user performs faster than the PT so monotonic PT subsequence $\{a_3, a_4\}$ is aligned with one single user frame b_4 . In type 4 the user's movement does not reach the required amplitude, so non-monotonic PT subsequence $\{a_{17}, a_{18}, a_{19}\}$ is aligned with one single user frame b_{21} . Type 5 represents the basic case where one PT frame is aligned with one user frame.

TABLE II
FIVE ALIGNMENT TYPES IN DTW

Type	Number of frames PT	Number of frames User	Monotonicity of subsequence	User's performance
1	>1	1	Monotonic	Too Fast
2	1	>1		Too Slow
3	1	>1	Non-Monotonic	Overdone
4	>1	1		Incomplete
5	1	1		Matches PT

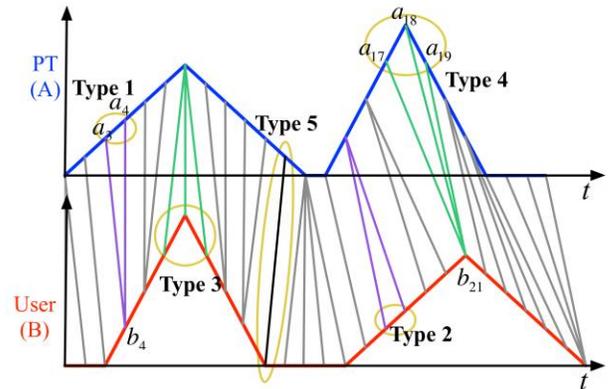


Fig. 8. Five alignment types in DTW: 1) The user moves faster. 2) The user moves slowly. 3) User's overdone motion. 4) User's incomplete motion. 5) Basic case where one PT frame is aligned with one user frame.

Next, the PT's motion data are considered as the ground truth and the user's error can be calculated by comparing each PT frame and the aligned user frame/frames. If there is only one single user frame b_j aligned with the PT frame a_i (i.e., type 5), the user's error in this frame can be computed as

$$e_{frame} = \|a_i - b_j\|. \quad (11)$$

However, if PT frame a_i is aligned with multiple user frames $B^* = \{b_j, b_{j+1}, \dots, b_{j+w-1}\}$, the difference between the two sequences will be counted several times according to (11). In this case we should revise (11) to count in the user error for only once based on the alignment types in Table II and Fig. 8. If B^* is monotonic (i.e., type 2), the user performs slower than the PT. For most physical therapy tasks, user's speed is not important. (Tasks for which speed is important are not discussed in this paper.) Only the average user error should be counted, and (11) can be revised as

$$\hat{e}_{frame} = \left\| a_i - \left(\frac{1}{w} \sum_{r=0}^{w-1} b_{j+r} \right) \right\|. \quad (12)$$

If B^* is non-monotonic (i.e., type 3) which represents the user's overdone movements, the largest user error needs to be counted, and (11) can be revised as

$$\tilde{e}_{frame} = \max_{0 \leq r \leq w-1} \|a_i - b_{j+r}\|. \quad (13)$$

For type 1 and 4 where multiple PT frames are aligned with one single user frame, user's error will be calculated separately for each PT frame according to (11). Based on the discussion above, the user's overall error on this criterion can be obtained by averaging the user's error for each PT frame.

2) Overall Score Estimation

In the previous section, we discussed how to calculate the user's error on each criterion. Combining them into a vector we can get the user's error vector e for the task. In this section, we will introduce how to transform the error vector e into a normalized overall score that indicates the user's overall performance for this task.

To obtain the score estimation model, a subjective study is needed where the proposed system calculates the error vector e and the physical therapist gives a true score s for each subject. Given the error vectors $\{e_1, e_2, \dots, e_i, \dots, e_N\}$ and the corresponding scores $\{s_1, s_2, \dots, s_i, \dots, s_N\}$ ($s_i \in [0, 10]$) for N samples, our goal is to find an optimal function $h(e)$ so that $s_i \approx h(e_i)$. Here we choose h to be linear and include constant 1 in e_i as the bias term. Thus h can be represented as

$$h(e) = \beta^T e. \quad (14)$$

We use linear regression [32] to estimate the optimal β^* as

$$\beta^* = (X^T X)^{-1} X^T y, \quad (15)$$

where

$$X = (e_1, e_2, \dots, e_N)^T, y = (s_1, s_2, \dots, s_N)^T. \quad (16)$$

From (15) the optimal parameter β^* can be calculated from all the scores given by the physical therapist and error vectors in the training set. Then this optimal function $h(e)$ can be used to estimate the overall score for any new user performance.

D. Real-Time Guidance and Satisfactory Score

In order to help the user improve performance accuracy, we propose a replay system which highlights the user's error and provides visual and textual guidance for the user. Fig. 9 shows a screenshot of the guidance system for the leg lift task. The overall score for the user's performance is shown on the upper left corner of the screen. Two avatars replay two views of the user's movements, with the view angles determined by the task to better show the user's error. In Fig. 9, the left avatar shows the side view and the right avatar shows the mirrored view. For each gesture, the user's motion data on each criterion are compared with the corresponding PT's motion data. If the user's error on a criterion is above the error threshold defined by the physical therapist (see Table I), the guidance video will be slowed down, and visual/textual guidance is provided for the user to calibrate his/her movements.

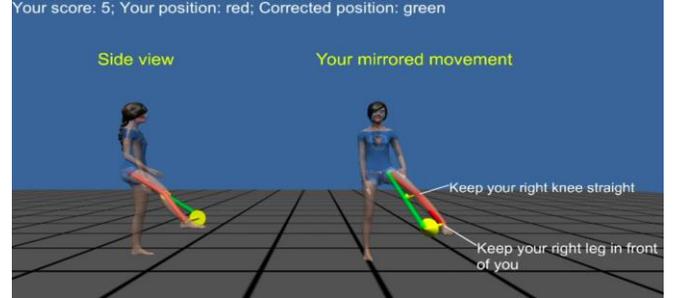


Fig. 9. Examples of textual and visual guidance in the leg lift task. Left avatar: side view. Right Avatar: mirrored view. User's incorrect body parts: red. Corrected position: green. Textual information is placed beside the body.

Visual guidance uses colored cylinders to label the user's incorrect body positions and the correct positions. Incorrect body positions are rendered in red and the corrected positions are rendered in green so user can see the clear difference. In addition, directional arrows rendered in yellow will give further guidance on how to correct this movement. Textual guidance is provided beside the corresponding body parts to instruct the user. There are two types of textual guidance: qualitative and quantitative textual guidance. Qualitative textual guidance gives only general instructions on how to calibrate incorrect motion (e.g., "bring your right leg higher"), while quantitative textual guidance provides detailed instructions on the quantitative error (e.g., "bring your right leg higher by 20 degrees"). Quantitative guidance is important for the user to make right calibrations and avoid over

corrections. However, when textual guidance is provided together with visual guidance, qualitative textual guidance may be sufficient since visual guidance already gives the user intuitive instructions about the quantitative error. To determine which kind of guidance is most helpful for the user, we have conducted subjective tests, whose results are shown in Section V-D.

In addition, there are multiple choices for the timing of providing guidance. For example, 1) concurrent guidance when the user is learning the task, or 2) knowledge of result, i.e., guidance after the user has done the whole training task, and 3) post-gesture guidance after the user finishes each gesture. Concurrent guidance is hard to achieve since the data alignment approach cannot be applied in hard real time. Besides, concurrent guidance may be overwhelming for the user. Too many instructions in training may cause user's failure in following the task. Guidance after the entire task is not real-time and cannot provide timely guidance for the user. Besides, for some tasks that include multiple different gestures and last several minutes, the user may have forgotten his/her performance on the first few gestures, which may cause the guidance to be ineffective. Post-gesture guidance can be considered soft real-time and can make it easier for the user to utilize the guidance. Moreover, post-gesture guidance can be fully personalized depending on the user's performance. For good user performance, no guidance is needed and the user can continue his/her training. When the user makes some errors in a gesture, he/she will receive timely guidance after this gesture.

Hence, we believe that post-gesture guidance is the most helpful in the proposed system. Real-time gesture segmentation has been achieved by the proposed GB-DTW-A algorithm. To determine whether to provide guidance or continue training, a satisfactory score is set by the physical therapist (which will be discussed in Section V). Scores above the satisfactory score means that the user passes the gesture and can progress to the next gesture. Otherwise, the system will pause the training and provide guidance for this gesture.

V. EXPERIMENTAL RESULTS

We conducted experiments based on the testbed (shown in Fig. 10) we have developed to emulate the system architecture in Fig. 1. The cloud server is running on a desktop with a quad core 3.1GHz CPU and 8GB RAM, and the user device is a laptop PC with a dual core 2.5GHz CPU and 4GB RAM. The network connection between the cloud server and the mobile laptop is emulated using a network emulator (Linktropy [33]), which can be programmed to emulate different wireless network profiles. All the experiments were conducted with the assistance of a licensed physical therapist who specializes in movement disorder population with a background in orthopedics and fitness.

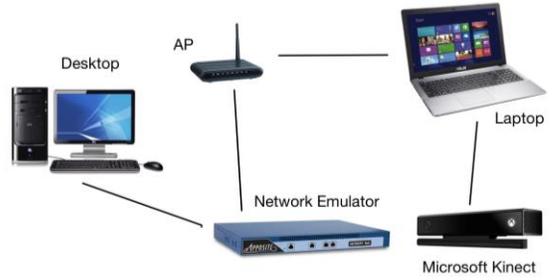


Fig. 10. Experiment testbed.

A. Experiments to Validate Data Alignment Approach

To validate the proposed data alignment and gesture segmentation approach, the tested task is shoulder abduction and adduction (shown in Fig. 2(a), criteria and motion features are shown in Table III) with different target heights for five times. The PT's motion data for the five gestures are shown as the blue curve in Fig. 11. Only the left shoulder angle is shown here for simplicity.

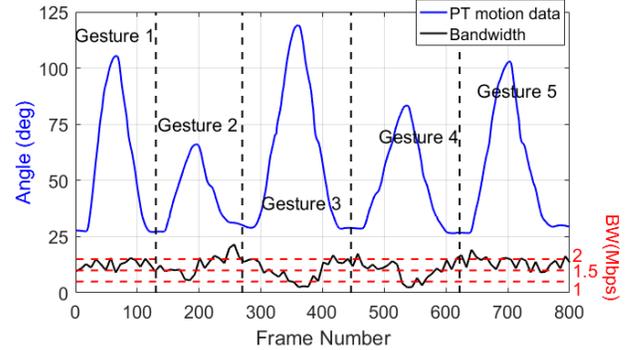


Fig. 11. PT's motion data (i.e., left shoulder angle) and the bandwidth profile.

Four users (User A, B, C and D) with different motion abilities were invited as subjects in the experiment. They tried to follow the PT's movements by watching the training video which was transmitted through the network emulator to the laptop. Each user was tested under ideal network condition (without any bandwidth constraint) and non-ideal network condition (limited by a bandwidth profile to simulate the downlink network). The bandwidth profile is shown as the black solid curve in Fig. 11 and it was repeated for each user using the network emulator. It can be observed that the bandwidth is relatively lower at the third and fourth gestures. Then we use three different techniques: 1) tradition method of MCC, 2) classical DTW on the entire motion sequences, and 3) GB-DTW-A, to align the motion sequences of the PT and the user. For the GB-DTW-A algorithm, the double thresholds $\{\tau_S, \tau_L\}$ are set as $\{0.1, 0.5\}$ and $\tau_M = 0.5$, $r = 20$, $\delta = 5\%$. The alignment results of user A are shown in Fig. 12. In each figure, we plot the motion data of the PT and the user, with the x -axis showing the frame number and the y -axis showing the tested shoulder angle. The vertical dashed lines in GB-DTW-A show the gesture segmentation results. In the two DTW algorithms, when multiple frames in one sequence are aligned with one single frame in the other sequence, the single frame is repeated

TABLE III

MOTION FEATURES AND CRITERIA OF SHOULDER ABDUCTION AND ADDUCTION, LEG LIFT AND JUMPING JACK

Task	Feature Type	Criteria	Feature
Shoulder abduction and adduction	Time-varying ($N_v = 1$)	“Raise the arm to the required height”	Angle between the arm and the vertical direction: set by the PT (e.g., 90°)
	Constraint ($N_c = 1$)	“Keep the elbow straight”	Angle between the upper arm and lower arm: 180°
Leg lift (right)	Time-varying ($N_v = 1$)	“Lift right leg to the required height”	Angle between the right leg and the vertical direction: set by the PT (e.g., 60°)
	Constraint ($N_c = 4$)	“Keep the trunk upright”	Angle between the trunk and the vertical direction: 0°
		“Keep pelvis level”	Angle between the pelvis and the horizontal direction: 0°
		“Knee right knee straight”	Angle between the right thigh and shank: 180°
Jumping jack	Time-varying ($N_v = 2$)	“Raise left arm beyond the head”	Angle between left arm and the vertical direction: 120°
	Constraint ($N_c = 5$)	“Raise right arm beyond the head”	Angle between right arm and the vertical direction: 120°
		“Keep left and right arm symmetrical”	Difference between the two time-varying features: 0°
		“Keep left arm aligned with the trunk”	Angle between the left arm and the body plane: 0°
		“Keep right arm aligned with the trunk”	Angle between the right arm and the body plane: 0°
		“Keep left elbow straight”	Angle between the left upper arm and lower arm: 180°
“Keep right elbow straight”	Angle between the right upper arm and lower arm: 180°		

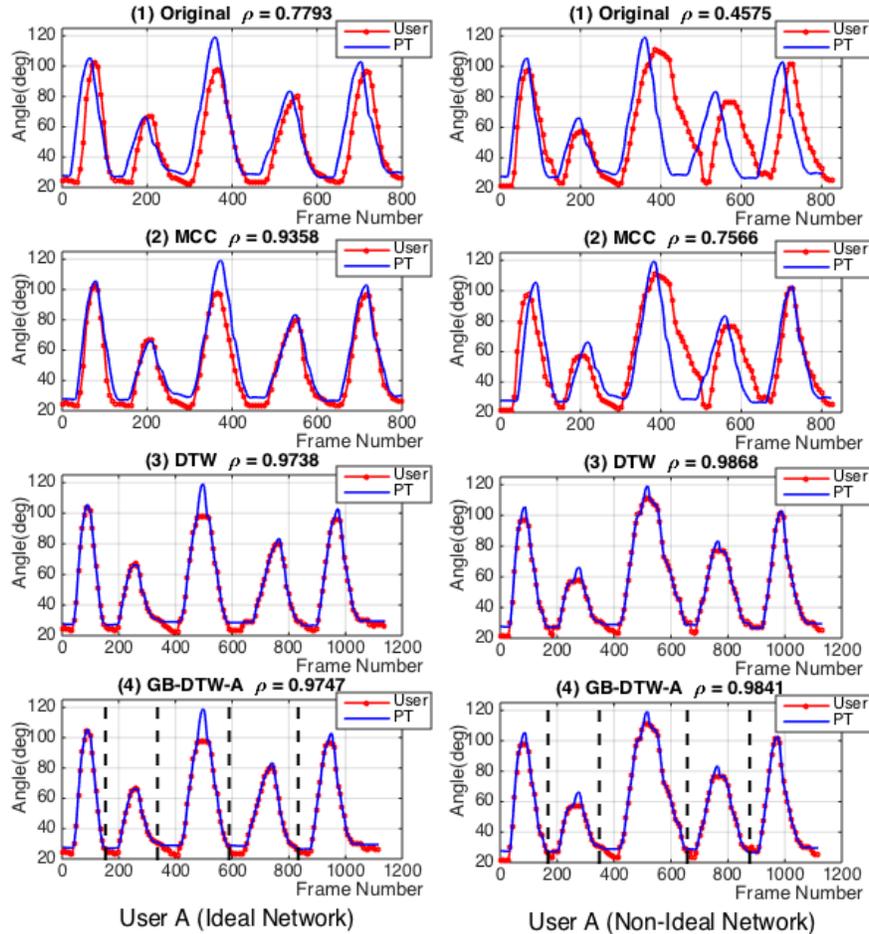


Fig. 12. Data alignment results for User A under ideal and non-ideal network conditions. (1) Original misaligned motion sequences of the PT and the user. (2) Aligned sequences using MCC. (3) Aligned sequences using classical DTW on the entire sequence. (4) Aligned sequences using GB-DTW-A and gesture segmentation.

for several times to show the alignment results. From Fig. 12 we can see that the user performs worse with fluctuating bandwidth than ideal network condition due to the network delay. Especially at the third and fourth gestures when bandwidth is limited, he/she cannot follow the PT and performs more slowly. To quantify the alignment results, we calculate the correlation coefficient ρ of the aligned sequences x and y in each method as

$$\rho = \frac{E[(x - \bar{x})(y - \bar{y})]}{\sqrt{\sigma_x^2 \sigma_y^2}}, \quad (17)$$

where \bar{x}, \bar{y} are the means of x, y and σ_x^2, σ_y^2 are the variances. High correlation coefficient indicates that the two sequences are aligned better. The correlation coefficients for each user using different methods are shown in Table IV. Comparing the three methods, it can be concluded that when the user follows the PT quite well and there is only time shift delay, the traditional method of MCC gives high correlation coefficients ($\rho > 0.85$). However, when the network condition is not ideal and therefore the training video is delayed, or when the user cannot follow the PT due to his/her motion ability, the user's motion data are distorted. In this case the two DTW algorithms perform much better ($\rho > 0.95$) than MCC ($\rho < 0.80$). For DTW and GB-DTW-A, their alignment results are quite close and both of their correlation coefficients are more than 0.95. Fig. 13 shows the running time of DTW and GB-DTW-A on the four users under ideal and non-ideal network conditions. We can see that GB-DTW-A needs significantly less time compared with DTW to align the two sequences, which validates our deduction in (10). Therefore, the proposed GB-DTW-A outperforms other alignment methods as well as enable real-time guidance with reduced computation complexity.

TABLE IV
CORRELATION COEFFICIENTS FOR USER A, B, C, AND D USING DIFFERENT ALIGNMENT METHODS UNDER IDEAL AND NON-IDEAL NETWORK CONDITIONS

User	Network Condition	Original	MCC	DTW	GB-DTW-A
A	Ideal	0.7793	0.9358	0.9738	0.9747
	Non-Ideal	0.4575	0.7566	0.9868	0.9841
B	Ideal	0.7824	0.9578	0.9741	0.9753
	Non-Ideal	0.4726	0.6104	0.9811	0.9827
C	Ideal	0.6388	0.8766	0.9654	0.9649
	Non-Ideal	0.1036	0.6351	0.9888	0.9729
D	Ideal	0.6190	0.9302	0.9752	0.9761
	Non-Ideal	-0.0944	0.7115	0.9851	0.9851

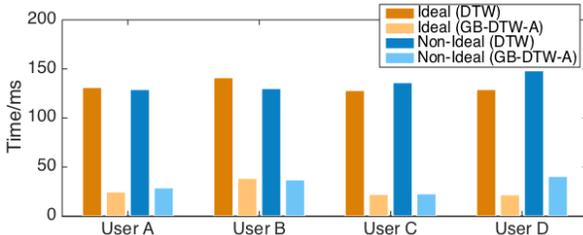


Fig. 13. Running time of DTW and GB-DTW-A under ideal and non-ideal network conditions.

B. Experiments to Compare GB-DTW0 and GB-DTW-A

Wireless networks can be associated with significant jitter (variations in network delay). Jitter can exacerbate the motion data misalignment problem due to network delay, and challenge the performance of the data alignment and segmentation algorithms GB-DTW0 [13] and GB-DTW-A. Hence, we conduct experiments to compare the performance of the two algorithms in the presence of jitter. We emulate the condition where the user follows the PT accurately, but his/her motion sequence is affected by jitter. By using this “perfect” user, the motion data misalignment is completely caused by jitter. Therefore, the effectiveness of the two algorithms can be tested by checking whether they can achieve “perfect” alignment result (correlation coefficient close to 1). In our experiments, the motion sequence of the “perfect” user is created by delaying each frame of the original PT's motion sequence shown in Fig. 11 by Δt . (Frames are not reordered even if subjected to differing delays.) Δt follows a positive truncated normal distribution (i.e., $\Delta t \sim |N(0, \sigma^2)|$), and the mean of Δt is

$$\mu_{\Delta t} = \int_0^{\infty} \Delta t \cdot \frac{2}{\sqrt{2\pi}\sigma} e^{-\frac{\Delta t^2}{2\sigma^2}} d\Delta t = \sqrt{\frac{2}{\pi}} \sigma. \quad (18)$$

$\mu_{\Delta t}$ is proportional to the standard deviation σ . Larger $\mu_{\Delta t}$ represents higher delay and jitter in the wireless network. In the experiments, $\mu_{\Delta t}$ ranges from 0s to 8s. For each value of $\mu_{\Delta t}$, experiments are repeated for ten times and the average is calculated. For GB-DTW0 and GB-DTW-A, we calculate the following four indexes.

Correlation Coefficient (CC): see (17).

User Error (UE): user's average error in each frame. In the shoulder abduction and adduction task, user error is in degrees since the motion feature is the shoulder angle.

Segmentation Error (SE): error between the detected endpoint and the true endpoint of the user's gesture.

Segmentation Delay (SD): delay between the true endpoint of the user's gesture and the time when the segmentation is completed.

Results are shown in Fig. 14, with the x-axis showing $\mu_{\Delta t}$ and y-axis showing CC, UE, SE, and SD in the four sub-figures respectively. Since each user motion sequence contains only network delay, the user's performance can be considered “perfect” and thus CC should be close to 1 and UE should be close to 0. Smaller SE indicates more accurate segmentation and smaller SD means more real-time segmentation. From Fig. 14 it can be concluded that, when the jitter is low (i.e., $\mu_{\Delta t} < 2s$), both GB-DTW0 and GB-DTW-A achieve good segmentation and alignment results. Note that the SD result of GB-DTW0 is always larger than 20 frames because Condition 3 is always used to check r frames following the gesture endpoint. When jitter is higher (i.e., $\mu_{\Delta t} > 4s$), GB-DTW-A shows superiority over GB-DTW0, especially in maintaining low SE and SD. The average number of CC, UE, SE, and SD for GB-DTW0 and GB-DTW-A are shown in Table V. We can

observe significant improvements achieved by the new algorithm GB-DTW-A compared to GB-DTW0 [13], especially in reducing estimation of user error (lower UE), enhancing segmentation accuracy (lower SE), and making segmentation real-time (lower SD). Note that the segmentation delay (SD) achieved by the new algorithm GB-DTW-A is only 11 frames on average, compared to an average of 39 frames for GB-DTW0, and never higher than 40 frames. The low SD numbers achieved by GB-DTW-A validates that the computation complexity of GB-DTW-A is close to $O(mn/g)$ in most cases, and since it never has to search till the end of the user sequence, it shows that it never reaches the worst-case computation complexity of $O(mn)$ (section IV-B).

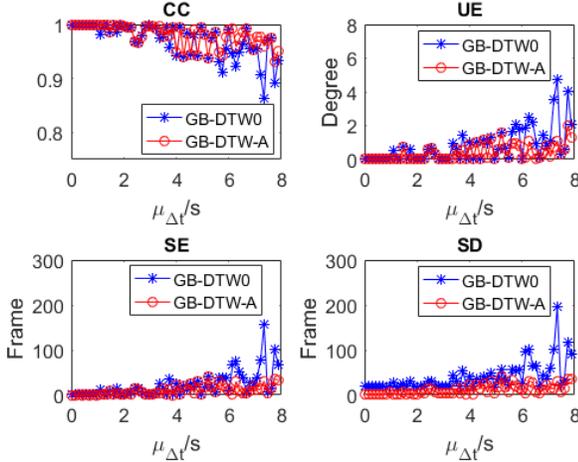


Fig. 14. Comparison between GB-DTW0 and GB-DTW-A. The four sub-figures show results of correlation coefficient (CC), user error (UE), segmentation error (SE), and segmentation delay (SD).

	CC	UE (degree)	SE (frame)	SD (frame)
GB-DTW0	0.97	0.78	21	39
GB-DTW-A	0.98	0.38	10	11
Improvement	0.95%	50.1%	54.1%	71.2%

C. Experiments to Estimate Overall User Score

As discussed in Section IV-C, the optimal function $h(e)$ for each task can be estimated by applying linear regression on training samples. In this experiment, the tested tasks are leg lift and jumping jack which are shown in Fig. 15. Motion features and criteria for each task are shown in Table III.

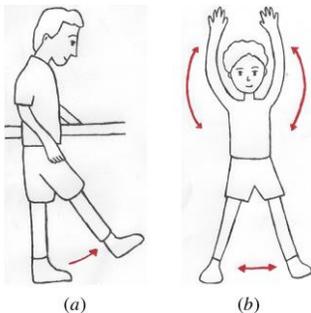


Fig. 15. (a) Leg lift. (b) Jumping jack.

In the experiment 10 subjects (aged 18~30, 6 males, 4 females) used the proposed training system to perform leg lift and jumping jack for several times. For each performance of each subject, the physical therapist gave an evaluation score $s \in [0,10]$. In the meantime, the proposed training system captured the subject's movements, processed the motion data and calculated an error vector e . 60 samples were gathered for each task.

All the samples are randomly divided into a training set (including 42 samples) and a test set (including 18 samples). For the training set, Equation (15) is used to train the samples and calculate $h(e)$. Then we apply the optimal function $h(e)$ on the test set. The results are shown in Fig. 16, with the x -axis showing the real score s_{PT} given by the physical therapist and the y -axis showing the estimated score $s_{estimated}$ using $h(e)$. The mean absolute error (MAE) between s_{PT} and $s_{estimated}$ is calculated and shown in Fig. 16. Samples on the diagonal line $s_{PT} = s_{estimated}$ means that the estimated score is the same as the real score without any error. The two dotted lines $s_{PT} = s_{estimated} \pm 1$ define the diagonal area for which the estimation error is below 1. (We choose 1 as the error threshold since most scores given by the physical therapist are integers, for which 1 is the minimum error.) We can see that most of the test samples lie in the diagonal area, which means that the evaluation models are accurate. Besides, using $h(e)$ to evaluate the patients is superior because the intra-rater reliability [34] of a human performing movement analysis without any analytical tools besides eye site shows increased variability. By utilizing the system to analyze the movements there is a more uniform scoring and increased intra-rater reliability.

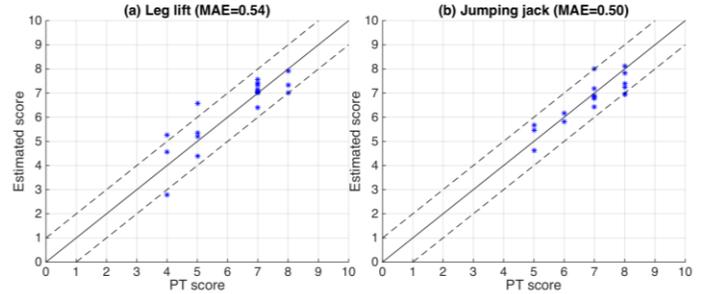


Fig. 16. Estimated score vs. PT real score and the mean absolute error (MAE) for (a) leg lift and (b) jumping jack.

D. Effectiveness of Visual and Textual Guidance

As discussed in Section IV-D, visual and textual guidance can be provided after each gesture according to the user's performance. The satisfactory score is set as 7 by the physical therapist in order to allow for some intrinsic error correction, which would allow for increased learning of the task. If the threshold is set too low, the patients would obtain a passing score too easily and not have the correct amount of feedback to properly correct the deficits in his/her movements. If the score is set too high, it might discourage the patients from trying their best and create a negative mindset, resulting in a reduction in retention.

To validate the effectiveness of the guidance system, we conducted another subjective test to compare four types of guidance: 1) no guidance (N), 2) visual guidance (V), 3) quantitative textual guidance (T), 4) visual and qualitative textual guidance (VT). There are two alternative ways to design the subjective test. The first one is having each user try four different tasks with equal difficulty level, with each task associated with one type of guidance. The four tasks should be completely different, otherwise the user’s ability may improve after he/she tries one task which will impact his/her performance of the next task and hence our evaluation of the effectiveness of the associated guidance. The other way is dividing all the subjects into four groups, with equal average ability in each group. People in different groups practice the same task but are provided with different types of guidance. After consultation with the physical therapist and multiple attempts of data capture, it was not clear if it is possible to have tasks which are significantly different from each other and yet have same quantifiable difficulty level, because of the tracking insufficiency of the Kinect sensor for some tasks (e.g., use of wheelchairs, occlusion problem). Hence we considered the first method to be not feasible, and instead decided to use the second method.

In the test, 28 subjects (aged 17 ~ 38, 14 males, 14 females) were invited to perform two training tasks (leg lift and jumping jack) using the proposed system. To ensure the same initial average score of each group, groups were assigned after the first attempt of each subject. Each subject performed each task four times and the average score of each group is calculated. Fig. 17 shows the average performance and 90% confidence intervals (black vertical lines) of each group, with each group represented by a different color. The red dotted curve shows the satisfactory score.

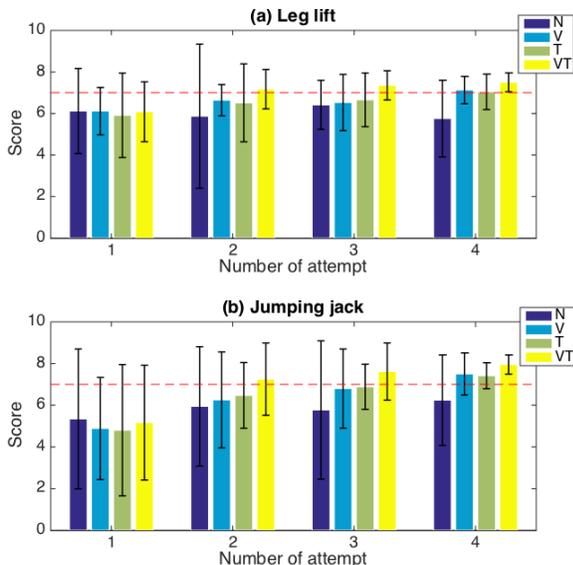


Fig. 17. Average score of each group with vertical lines showing 90% confidence interval. (a) Leg lift. (b) Jumping jack.

From Fig. 17 we can see that the average scores on the first attempt in each group are similar, which ensures similar initial

ability of each group. We also make the following important conclusions from the results. While scores for people in group N (without any kind of guidance) fluctuates with large confidence intervals, and may or may not reach the satisfactory score, using each type of real-time guidance helps the users improve performance, though with varying effectiveness. People in group V (who get visual guidance) and group T (who get quantitative textual guidance) reach the satisfactory score 7 after the fourth attempt. On the other hand, the results show that the combination of visual and textual guidance is the most helpful: it helps users in group VT reach score 7 after only the second attempt.

E. Performance Validation Using Real Cloud Environment

To validate the performance of the proposed system on a real cloud environment, we implemented the system on Amazon Web Services (AWS) [35]. The experiment setup is the same as Fig. 10 except that the desktop and network emulator are replaced by AWS (and the real network from AWS to the user device). Specifically, we use AWS g2.2xlarge instance which provides access to one NVIDIA GRID GPU with 1,536 CUDA cores and 4GB of video memory. The CPU it provides is Intel XeonE5-2670 @2.60GHz with 15GB memory. The operating system we deploy is Windows_Server-2008-R2_SP1.

One of the concerns of having the system run on a real cloud environment is the possible impact of additional delay from the cloud to the user device. We tested the delay of the training and guidance videos under three different network conditions: 1) unloaded network (e.g., accessing our cloud-based system using home Wifi at midnight), 2) loaded network (e.g., accessing our cloud-based system using LTE network at 5pm), 3) loaded and noisy network (e.g., accessing our cloud-based system using public Wifi at 5pm). The histograms of the measured delay under each condition are shown in Fig. 18, with the x -axis showing the delay and the y -axis showing the frequency of each value (i.e., number of occurrences of the value).

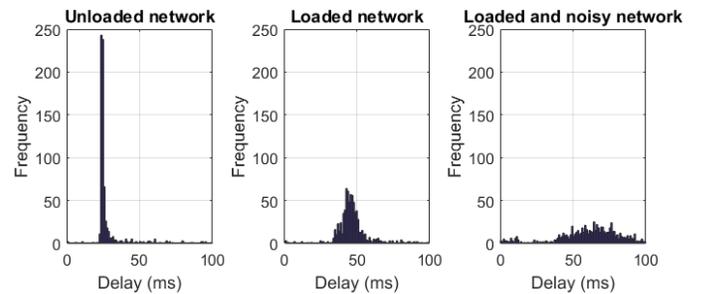


Fig. 18. Histogram of the measured delay of avatar video from cloud (AWS) to user device under unloaded, loaded, and loaded and noisy network conditions.

The mean and Standard Deviation (STD) of the measured delay are shown in Table VI. When the network is unloaded, the delay is under 30ms most of the time. When the network is loaded and noisy, the delay is increased significantly but still

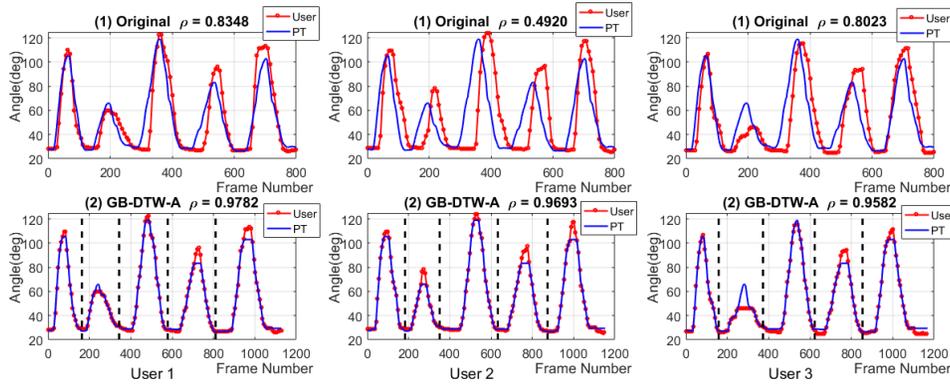


Fig. 19. Data alignment results for User 1, 2, 3 using AWS. (1) Original misaligned motion sequences of the PT and the user. (2) Aligned sequences using GB-DTW-A and gesture segmentation.

under 100ms, which means that the video streaming from the cloud to the user’s mobile device can be considered real-time in the system. Furthermore, we invited three new users to perform the shoulder abduction and adduction task using the proposed training system. The motion data alignment algorithm (i.e., GB-DTW, see Section IV-B) and the user performance evaluation algorithm (see Section IV-C) are implemented on AWS. Fig. 19 shows the motion data alignment results. We can see that the proposed GB-DTW algorithm still works well in aligning motion data and segmenting gestures in the real cloud environment. Table VII shows the running time of the alignment and evaluation algorithms on AWS. The running time of the two algorithms are under 20ms, again demonstrating their real-time nature. From all the above results, it can be concluded that the proposed system is able to provide real-time training and guidance for the user in a real cloud environment.

TABLE VI

MEAN AND STD OF DELAY FROM CLOUD TO USER DEVICE UNDER UNLOADED, LOADED, LOADED AND NOISY NETWORK CONDITIONS

	Unloaded	Loaded	Loaded and noisy
Mean (ms)	28.09	46.90	60.72
STD (ms)	11.85	11.19	20.99

TABLE VII

RUNNING TIME OF GB-DTW-A AND USER PERFORMANCE EVALUATION ALGORITHMS IN CLOUD (AWS)

Algorithm	User 1	User 2	User 3
GB-DTW-A (ms)	16.87	14.91	14.90
User performance evaluation (ms)	0.35	0.29	0.38

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a cloud-based physical therapy monitoring and guidance system that captures and evaluates the user’s performance automatically. It can also be applied to many other types of training applications, such as wellness and fitness training, and ergonomics training. To address the

motion data misalignment problem as well as enable real-time evaluation, we propose the GB-DTW-A algorithm to align the motion data and segment the user’s motion sequence into gestures in real time with reduced computation complexity. Experiments with multiple subjects using real network profiles show that the proposed method works better than other alignment techniques. Moreover, we provide results to demonstrate the accuracy and real-time performance of the proposed GB-DTW-A algorithm. Furthermore, the evaluation model for the user’s performance is trained based on subjective test and linear regression method. Testing results show that the evaluation model is able to provide an accurate score which is quite close to the real score given by the physical therapist for the user’s performance. Besides, the proposed guidance system can provide detailed visual and textual guidance, whose effectiveness has been validated in subjective test. Experiments using real cloud environment AWS show that the proposed system can provide real-time training and guidance for the user.

In the future, we may incorporate other kinds of sensors, like pressure sensors and epidermal sensors. Besides, Kinect can lead to inaccurate and unstable skeleton tracking, especially when tracking complex movements or patients with walkers and wheelchairs. Hence we would like to use multiple cameras or incorporate other motion capture sensors to improve the skeleton capture accuracy. Moreover, setting uniform criteria for different patients may cause injury or over corrections. Thus we would like to make the criteria of each task more adaptive and personalized for patients according to their health conditions. Furthermore, we will explore more about the design of guidance. Currently the proposed visual and textual guidance are proved useful for the user to improve performance, and the combination of visual and textual guidance is the most helpful. However, many other issues need to be considered to improve the effectiveness of guidance, e.g., are there other types of guidance which may be more effective for certain types of patients, what is the proper frequency to provide guidance, and how much guidance might be right as opposed to being overwhelming for the user. All of these issues need to be considered and explored in our future work.

REFERENCES

- [1] L. Catarinucci, et al., "An IoT-aware architecture for smart healthcare systems," *IEEE Internet of Things Journal* 2.6 (2015): 515-526.
- [2] Z. Yang, et al., "An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare," *Journal of medical systems* 40.12 (2016): 286.
- [3] K. Aziz, et al., "Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies," *Big Data and Smart City (ICBDSC'16)*, Muscat, March, 2016.
- [4] Z. Ali, G. Muhammad, and M. F. Alhamid, "An Automatic Health Monitoring System for Patients Suffering From Voice Complications in Smart Cities," *IEEE Access* 5 (2017): 3900-3908.
- [5] P. Choden, et al., "Volatile urine biomarkers detection in type II diabetes towards use as smart healthcare application," *Knowledge and Smart Technology (KST'17)*, Chonburi, February, 2017.
- [6] Kinect. [Online]. Available: www.xbox.com/en-US/kinect
- [7] D. Jack, et al., "Virtual reality-enhanced stroke rehabilitation," *Neural Systems and Rehabilitation Engineering*, IEEE Transactions on, 9.3 (2001): 308-318.
- [8] A. Mirelman, B. L. Patriiti, P. Bonato, and J. E. Deutsch, "Effects of virtual reality training on gait biomechanics of individuals post-stroke," *Gait & posture*, 31.4 (2010): 433-437.
- [9] The U.S. Mobile App Report by comScore. [Online]. Available: <http://www.comscore.com/Insights/Presentations-and-Whitepapers/2016/The-2016-US-Mobile-App-Report>.
- [10] M. T. Nkosi, and F. Mekuria, "Cloud computing for enhanced mobile health applications," *Cloud Computing Technology and Science (CloudCom'10)*, Indianapolis, December, 2010.
- [11] Y. Lu, Y. Liu, and S. Dey, "Cloud mobile 3D display gaming user experience modeling and optimization by asymmetric graphics rendering," *IEEE Journal of Selected Topics in Signal Processing* 9.3 (2015): 517-532.
- [12] Unity. [Online]. Available: <https://unity3d.com/>
- [13] W. Wei, Y. Lu, C. Printz and S. Dey, "Motion Data Alignment and Real-Time Guidance in Cloud-Based Virtual Training System," in *Proc. of Wireless Health (WH'15)*, Bethesda, Oct. 2015.
- [14] S. Ananthanarayan, et al., "Pt Viz: towards a wearable device for visualizing knee rehabilitation exercises," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*, Paris, April, 2013.
- [15] C. Y. Chang, et al., "Towards pervasive physical rehabilitation using Microsoft Kinect," *Pervasive Computing Technologies for Healthcare (PervasiveHealth'12)*, San Diego, May, 2012.
- [16] B. Lange, et al., "Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor," *Engineering in Medicine and Biology Society (EMBC'11)*, Boston, September, 2011.
- [17] Y. J. Chang, S. F. Chen, and J. D. Huang, "A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities," *Research in developmental disabilities* 32.6 (2011): 2566-2570.
- [18] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice, "YouMove: enhancing movement training with an augmented reality mirror," *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST'13)*, St Andrews, October, 2013.
- [19] D. S. Alexiadis, et al., "Evaluating a dancer's performance using kinect-based skeleton tracking," in *Proc. of the 19th ACM international conference on Multimedia (MM'11)*, Scottsdale, November, 2011.
- [20] A. Yurtman, and B. Barshan, "Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units," *Information Sciences and Systems (SIU'14)*, Trabzon, April, 2014.
- [21] O. Bau, and W. E. Mackay, "OctoPocus: a dynamic guide for learning gesture-based command sets," *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST'08)*, Monterey, October, 2008.
- [22] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor, "ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures," *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS'09)*, Banff, November, 2009.
- [23] R. Sodhi, H. Benko, and A. Wilson, "LightGuide: projected visualizations for hand movement guidance," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*, Austin, May, 2012.
- [24] J. Doyle, C. Bailey, B. Dromey, and C. N. Scanail, "BASE-An interactive technology solution to deliver balance and strength exercises to older adults," *2010 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth'10)*, Munich, March, 2010.
- [25] R. Tang, X. D. Yang, S. Bateman, J. Jorge, and A. Tang, "Physio@ Home: Exploring visual guidance and feedback techniques for physiotherapy exercises," *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*, Seoul, April, 2015.
- [26] The Microsoft documentation for Kinect 2.0. [Online]. Available: <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>
- [27] S. Dey, Y. Liu, S. Wang, and Y. Lu, "Addressing response time of cloud-based mobile applications," *Proceedings of the first international workshop on Mobile cloud computing & networking*, ACM, 2013.
- [28] D. J. Berndt, and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," *KDD workshop*, Vol. 10. No. 16. 1994.
- [29] M. Müller, "Information retrieval for music and motion," Vol. 2. Heidelberg: Springer, 2007.
- [30] K. Kahol, P. Tripathi, and S. Panchanathan, "Automated gesture segmentation from dance sequences," *Proceedings of the Sixth IEEE International conference on Automatic Face and Gesture Recognition (FGR'04)*, Seoul, May, 2004.
- [31] D. Kim, J. Song, and D. Kim, "Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs," *Pattern recognition*, 40.11 (2007): 3012-3026.
- [32] G. A. Seber, and A. J. Lee, "Linear regression analysis," Vol. 936. John Wiley & Sons, 2012.
- [33] Linktropy. [Online]. Available: <http://www.apposite-tech.com/products/>
- [34] K. L. Gwet, "Intrarater reliability," *Wiley encyclopedia of clinical trials*, 2008.
- [35] Amazon Web Services. [Online]. Available: <https://aws.amazon.com/>