Motion Data Alignment and Real-Time Guidance in Cloud-Based Virtual Training System

Wenchuan Wei[1], Yao Lu[1], Catherine D. Printz[2], Sujit Dey[1]
[1]Mobile Systems Design Lab, Dept. of Electrical and Computer Engineering
[2]Department of Rehabilitation Services, Thornton Hospital
University of California, San Diego
Email: {wenchuan, luyao, caprintz}@ucsd.edu, dey@ece.ucsd.edu

ABSTRACT
In this paper, by making use of virtual reality technology, motion sensors and cloud computing platform, we propose a cloud-based virtual training system for physical therapy, which enables a user to be trained by following a pre-recorded avatar instructor and getting real-time guidance using mobile device through wireless network. To evaluate the user's performance, we compare the motion data of the user and the pre-recoded avatar instructor. However, human reaction delay and network delay cause the data misalignment problem in the proposed cloud-based virtual training system. To align the motion data and evaluate the user's performance, we use Dynamic Time Warping (DTW) to calculate the similarity between the two sequences. Moreover, we propose a variant of the DTW algorithm we term Gesture-Based Dynamic Time Warping (GB-DTW) which segments the whole motion sequence and provides evaluation score for each gesture in real time. Experiments with multiple subjects under real network condition show that the proposed GB-DTW algorithm performs much better than other evaluation methods. To help the user calibrate his movements, the proposed system also provides visual and textual guidance for the user.

Categories & Subject Descriptors
J.3 [**Computer Applications**]: Life and Medical Sciences, *Health*

General Terms
Algorithms, Design, Experimentation, Performance

Keywords
Virtual Reality, Physical Therapy, Dynamic Time Warping, Gesture Segmentation, Real-Time Guidance

1. INTRODUCTION
In recent years, with the development of virtual reality technologies and evolution of motion capture sensors such as Microsoft Kinect [1], more and more user-motion based highly interactive applications are being developed, including motion sensing games [2], virtual training systems [3, 4], etc. In the meantime, mobile devices overtake PC in people's daily life. In June 2014, mobile application and browser accounted for 60% of digital media time spent in the United States [17]. In addition, cloud computing is being used as an alternative approach for mobile health applications [13], computer games [16], etc., to makes 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 video are processed and rendered on the cloud, which makes it superior to local processing on desktop computer for saving development effort for multiple device platforms and ubiquitous access from any device. 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 more usable. In this paper, we combine 1) virtual reality technology, 2) motion capture based on Microsoft Kinect, and 3) cloud computing for mobile devices, to propose a cloud-based real-time virtual training and guidance system, which enables a user to be trained by following a pre-recorded avatar instructor and getting real-time textual and visual guidance on a mobile device over a wireless network. In our proposed system, the avatar is rendered in the cloud and resulting video streamed to the user mobile device, while the user motion data are streamed from the motion sensors to the cloud for assessing accuracy in comparison with avatar motion and providing guidance to the user.

The proposed system has the ability to more effectively and efficiently train people for different types of physical therapy tasks like knee rehabilitation, shoulder stretches, etc. Although there exist other avatar-based training systems, our system provides real-time guidance rather than just providing scores, rendering our system unique. 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. Although the platform has the advantages as mentioned above, human reaction delay (delay by user to follow avatar instructions/motion) and mobile network delay (which may delay when the cloud rendered avatar video reaches the user device) may cause challenge for correctly calculating the accuracy of the user's movement compared to the avatar instructor's movement. 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 avatar instructor correctly. Therefore, we propose a Dynamic Time Warping (DTW) based algorithm to address the problem of motion data misalignment. We further apply a variant of the DTW algorithm we term Gesture-Based Dynamic Time Warping (GB-DTW) to segment the gestures among the whole motion sequence to enable real-time visual guidance to the user. We have implemented the proposed algorithms in a prototype avatar based real-time guidance system and conducted experiments using mobile network profiles. The experimental results show the performance advantage of our proposed method over other evaluation methods, and the feasibility of our proposed cloud-based mobile virtual training and guidance system.

Wireless Health '15, Oct 14-16, 2015, Bethesda, MD, USA

The rest of this paper is organized as follows: Section 2 reviews related work in improving automatic motion sensor based training systems. In Section 3, we introduce the architecture of the proposed cloud-based virtual training system and the data misalignment problem caused by the two kinds of delay. Section 4 proposes the GB-DTW algorithm to segment the whole sequence into gestures and its use in the proposed system to evaluate the performance of user as well as provide real-time textual and visual guidance. Section 5 presents the experimental results of motion data alignment and performance evaluation under real wireless network conditions. Section 6 concludes the paper and discusses future work.

2. RELATED WORK

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 perform the activities at home, in most cases following paper instructions and figures 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 professional physical therapists. To address this problem, some automatic training systems have been created to evaluate people's performance against standard or expected performance. In [4], the authors used the marker-based optical motion capture system Vicon and proved its effectiveness in gait analysis on subjects with hemiparesis caused by stroke. Furthermore, Microsoft Kinect sensor was proved of high accuracy and more convenient in detecting the human skeleton compared with wearable devices [5]. Authors in [11] developed a game-based rehabilitation system using Kinect for balance training. To evaluate the performance of the user's movement against the standard/expected movement, [6] used Maximum Cross Correlation (MCC) to compute 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. For two discrete-time signals f and g, their cross correlation $R_{f,g}(n)$ is given by

$$R_{f,g}(n) = \sum_{m=-\infty}^{\infty} f^*(m)g(m+n) \quad (1)$$

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

$$\tau = \arg\max_n \{R_{f,g}(n)\} \quad (2)$$

When the lengths of the two sequences are very close, shifting one sequence by the estimated delay τ can align them and their similarity can be calculated. However, this method calculates the overall delay for the entire sequence and cannot address the problem of variant human reaction delay and network delay, which will be discussed in Section 3.

In [7], DTW was used to detect and identify correct and incorrect executions of an exercise. However, the system in [7] was based on inconvenient wearable motion sensors and 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. In comparison, the proposed system in this paper does not need any pre-recorded error template to evaluate the performance and can provide detailed guidance for the user on how to improve the performance. Besides, all of the above techniques can only be applied offline when the entire motion sequence of the user is obtained, while the proposed system processes the motion data on the cloud and provides visual guidance in real time.

3. DATA MISALIGNMENT IN CLOUD-BASED VIRTUAL TRAINING SYSTEM

The architecture of the proposed cloud-based virtual training system is shown in Figure 1. An open-source character animation software platform, Smartbody [8], is used offline to pre-recode an avatar instructor's movements for a physical therapy exercise. During a user home training session, a cloud server uses Smartbody to render the avatar instructor for the exercise, with the rendered training video encoded and transmitted through a wireless network to the user mobile device. The user watches the decoded video on the mobile device and tries to follow it. Simultaneously, the user's movements are captured by Microsoft Kinect and uploaded to the cloud through the wireless network. On the cloud, motion data of the avatar instructor and user are compared and analyzed to determine accuracy. The results of accuracy are then processed by guidance logic followed by guidance rendering, and the guidance video is transmitted to the user device through the wireless network.

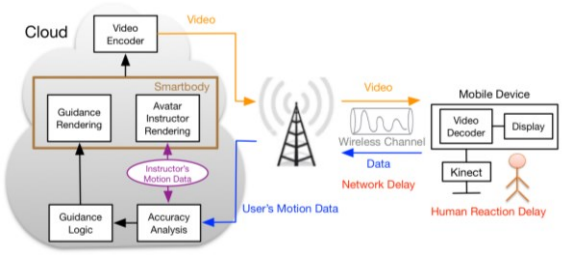


Figure 1. Cloud-based virtual training system

In the proposed system, Kinect captures 20 joints with *x, y, z* component for each joint. For a given exercise, some specific body parts might be deemed important. For frame i, we include joint coordinates of these important body parts as the feature vector f_i. Apart from joint positions, some other quantities that are derived from the joint coordinates, like joint angles, could also be included in f_i. The combination of the feature vector for each frame is the motion data $F = \{f_1, f_2, ..., f_m\}$ for the entire exercise.

Given the motion data of the avatar instructor and the user, we compute 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 delays in the system: human reaction delay and network delay. In the rest of this section, we will discuss the two kinds of delay, and discuss the problems the existing technique MCC [6] has in addressing the misalignment between the two sequences.

3.1 Human Reaction Delay

After seeing the movement of the avatar instructor on the screen, it may take the user some time to react to this movement and then follow it. We term the time period from when the avatar instructor starts the motion till the user starts the same motion as human reaction delay. However, for those training exercises including multiple separate gestures, the user's reaction delay might be different for these gestures. Here we define a gesture as a subsequence that represents the meaningful action of some body parts, especially when these body parts move and then return to the initial position. For example, raising one's hand and then putting it down can be considered as a gesture. Gestures in a

training exercise are segmented offline by physical therapist. Figure 2 shows the motion data of the avatar instructor and the user in an exercise of three gestures. For each gesture, the user follows the avatar instructor to laterally move his left arm from the solid position to the dotted position, and then return to the solid position. The corresponding motion data is the angle of the left shoulder θ. If there is only human reaction delay, we can assume that the user performs each gesture with time delay τ_1, τ_2 and τ_3 ($\tau_1 \neq \tau_2 \neq \tau_3$) but the time length of his gesture is close to that of the avatar instructor, i.e., $L_1' \approx L_1$, $L_2' \approx L_2$ and $L_3' \approx L_3$, where L_i and L_i' are the time length needed by the avatar instructor and user for gesture i respectively.

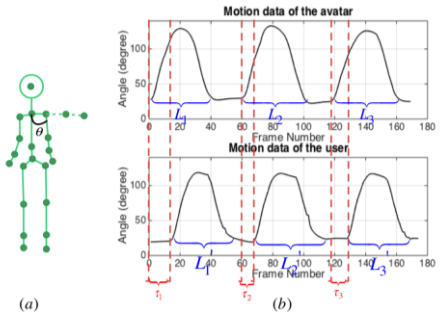


Figure 2. (a) Gesture of laterally moving one's left arm from the solid position to the dotted position and returning to the solid position. (b) Motion data (i.e., left shoulder angle) of the avatar instructor and the user in an exercise of three gestures as (a) with only human reaction delay. Delay for each gesture is τ_1, τ_2, τ_3.

Since the user's reaction delay varies with different gestures, using MCC to estimate an overall delay of the entire exercise sequence is unreliable. It is necessary to calculate the delay for each gesture separately.

3.2 Network Delay

In the proposed cloud-based virtual training system, the training video is transmitted to the user's mobile device through wireless network, which will result in network delay. Delay due to wireless network may vary from time to time and is influenced by many factors, such as bandwidth and the network load. Under the influence of network delay, the user may not only perform later than the avatar instructor, but may also perform slower depending on the amount of network delay during a gesture. Figure 3 shows the same training exercise as Figure 2.

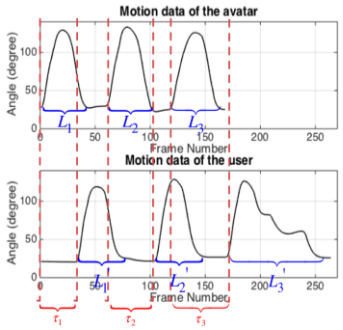


Figure 3. Motion data (i.e., left shoulder angle) of the avatar instructor and the user with both human reaction delay and network delay. The user performs the third gesture longer than the avatar instructor ($L_3' > L_3$) due to network delay.

With network delay, the time length of the user's gesture might be much longer than that of the avatar instructor's corresponding gesture. In this case, using MCC to shift the user's sequence by the estimated delay is unreliable. When the two sequences are different in length, a frame in the avatar instructor's motion sequence does not match the frame in the user's motion sequence that contains the corresponding movement of the user. To align the two sequences effectively and calculate their similarity, we need to rescale them on the time axis, i.e., extend or shrink the sequence horizontally, to match the total length of the other sequence.

4. DATA ALIGNMENT AND REAL-TIME GUIDANCE

To solve the data misalignment problem caused by human reaction delay and network delay, we propose a new data alignment method based on Dynamic Time Warping (DTW). In Section 4.1, we will introduce the principle of default DTW. Section 4.2 discusses the preprocessing of motion data when DTW is applied in the proposed system. In Section 4.3, Gesture-Based Dynamic Time Warping (GB-DTW) is proposed to segment gestures so that data alignment can be done in real time based on each gesture. Finally, we will describe how to rescale the motion sequences according to the alignment results of GB-DTW in Section 4.4, and the real-time guidance system in Section 4.5.

4.1 Dynamic Time Warping

DTW is a dynamic programming algorithm that is widely used in speech processing [9]. It measures the similarity of two sequences by calculating their minimum distance. Given sequences $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, an m×n distance matrix d is defined and $d(i, j)$ is the distance between a_i and b_j.

$$d(i,j) = \sqrt{|a_i - b_j|^2} \tag{3}$$

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

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

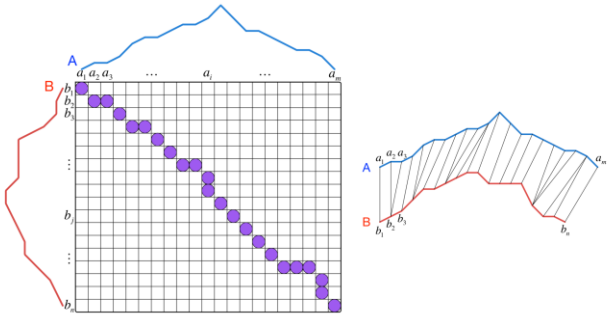
Under the three constraints, this path should start from $(1,1)$ and ends at (m,n). At each step, x_k and y_k will stay the same or increase by one.

To find this optimal path, an m×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 the following.

$$S(i,j) = d(i,j) + \min\begin{cases} S(i-1,j-1) \\ S(i,j-1) \\ S(i-1,j) \end{cases} \tag{4}$$

$S(m, n)$ is defined as the DTW distance of the two sequences [10]; smaller DTW distance indicates that the two sequences are more similar. The corresponding path indicates the best way to align the

two sequences. In this way, the two sequences are rescaled on the time axis to best match with each other. Time complexity of the DTW method is $\Theta(mn)$. Figure 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 mapping path of A and B. Figure 4(b) shows the corresponding alignment method given by the optimal path in Figure 4(a). For example, a_1 is aligned with b_1; a_2 and a_3 are aligned with b_2. In speech recognition, the DTW distance is calculated from a tested speech sample and several templates, with the sample classified as the pattern with the minimum DTW distance.



(a) (b)

Figure 4. (a) Warping path of DTW on sequence *A* and *B*. (b) Alignment result of *A* and *B*.

4.2 Data Preprocessing and Alignment

In Section 4.1, we introduce the principle of default DTW. For the data misalignment problem caused by human reaction delay and network delay, DTW can be used to rescale the two sequences on the time axis to align them. However, directly applying DTW on two sequences to evaluate their similarity is unreliable because the absolute amplitude of data may have influence on the optimal path and therefore the alignment result. We give an example to explain this problem. For two sequences $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, if we apply DTW on them, the alignment result is not expected to change if a constant c is added to B. However, when computing the new distance matrix of A and $B' = B + c$, (3) becomes

$$\begin{aligned} d'(i,j) &= \sqrt{|a_i - b_j'|^2} \\ &= \sqrt{|a_i - (b_j + c)|^2} \\ &\neq d(i,j) + c \end{aligned} \tag{5}$$

Therefore, the new distance matrix d' is different from d not only for the constant c. The relative size of elements in d is changed. Consequently the choice in (4) at each step might be different and $S' \neq S + c$. So B' is aligned with A in a different way.

To solve this problem, we propose to preprocess the data before applying DTW by aligning the two starting points a_1 and b_1 as (6).

$$B' = B + (a_1 - b_1) \tag{6}$$

Applying DTW on A and B', we can obtain the optimal path P^* and the DTW distance $S'(m,n)$ for A and B'

$$S'(m,n) = \sum_{(i,j)\in P^*} \sqrt{|a_i - b_j'|^2} \tag{7}$$

so the DTW distance $S(m,n)$ between the original data A and B *is*

$$S(m,n) = \sum_{(i,j)\in P^*} \sqrt{|a_i - b_j|^2} \tag{8}$$

In this way, we can first find out the optimal path P^* using the preprocessed data A and B', and then calculate the DTW distance for the original data A and B. (For convenience, in the following part of this paper, we consider the situation where the motion data have been preprocessed.)

Since the DTW distance $S(m,n)$ is a similarity measurement for the two sequences, we normalize $S(m,n)$ to 0~100 as evaluation score for the user. Smaller $S(m,n)$ represents higher score and indicates that the two sequences are more similar and the user performs better.

4.3 Real-Time Gesture Segmentation Based on DTW

In a physical therapy session using the proposed system, there are multiple ways to provide guidance to the user to help him calibrate his movements. For example, an entire replay of the movements that the user has performed together with the avatar instructor's movements can be provided after the user has done the whole training set (~several minutes). This can be classified as a non-real time feedback. Alternatively, we can provide feedback after the user finishes each gesture (~a couple of seconds), which can be considered as a real-time feedback. We believe that real-time feedback after every gesture can make it easier for the user to utilize the guidance; hence, we will research and address the challenges to enable real-time feedback.

In order to segment a motion sequence into gestures, there has been numerous research in the literature including methods based on machine learning, signal processing, etc [14, 15]. In this work, since DTW is used, we further propose a variant of DTW called GB-DTW so that the computation of DTW can be re-used to solve the gesture segmentation problem. The details of GB-DTW are presented in the following. We assume that for a given physical therapy exercise, gestures in the avatar instructor's motion sequence have been predefined and segmented by the physical therapist. Suppose that $A_1 = \{a_1, a_2, \ldots, a_{m_1}\}$ is defined as the first gesture in the avatar instructor's sequence $A = \{a_1, a_2, \ldots, a_m\}$. Then we would like to use DTW to find the subsequence of the user's motion data that matches the avatar instructor's gesture A_1 best. In [10], a modified DTW algorithm called subsequence DTW is used to search for a subsequence inside a longer sequence that optimally fits the other shorter sequence. Here we suppose that the starting point of one gesture is straight after the endpoint of the last gesture, so we can fix the starting point of the subsequence as b_1. For the subsequence $\{b_1, b_2, \ldots, b_k\}$ (k = 2, 3, …, n) of the user, its DTW distance with the avatar's gesture A_1 is $S(m_1,k)$. The optimal endpoint n_1 of the user's gesture should be the frame that leads to the best match between the two subsequences and gives the minimum DTW distance

$$n_1 = \arg\min_k \{S(m_1,k)\} \tag{9}$$

Due to the existence of local minimum points, the endpoint of the user's gesture cannot be determined until we obtain the whole motion sequence of the user. In [10], the entire sequence $B = \{b_1, b_2, \ldots, b_n\}$ is searched to find out the global minimum point. This means that we need to run from k = 2 to $k = n$ and find out the global minimum point, which will need significant additional computation.

Here we propose a new approach to estimate the global minimum point without testing k from 2 to n. For the global minimum point n_1, we know that $B_1 = \{b_1, b_2, \ldots, b_{n_1}\}$ matches $A_1 = \{a_1, a_2, \ldots, a_{m_1}\}$ best. When the user completes one gesture, he or she may stay in

the end position for some frames, and the feature vector of these frames will be quite close to b_{n_1}. So if we test e more frames after n_1, it is likely that all of these following frames $\{b_{n_1+1}, b_{n_1+1}, \ldots, b_{n_1+e}\}$ will be aligned to a_{m_1}.

Based on the insight above, we propose the following method to estimate the global minimum point. For each frame k of the user, we calculate the similarity of current subsequence $\{b_1, b_2, \ldots, b_k\}$ and the avatar instructor's gesture $A_1 = \{a_1, a_2, \ldots, a_{m_1}\}$ and get the DTW distance $S(m_1,k)$. When k increases from 2, $S(m_1,k)$ keeps decreasing in the beginning. If $S(m_1, k+1) > S(m_1, k)$, frame k is a local minimum point. To determine whether it is the global minimum point, we continue testing e frames and record DTW distances $S_{true} = \{S(m_1,k+1), S(m_1,k+2),\ldots, S(m_1,k+e)\}$. In the meantime we compute the estimated DTW distances $S_{estimated} = \{S'(m_1,k+1), S'(m_1,k+2),\ldots, S'(m_1,k+e)\}$ for the case where all of the following frames $\{b_{k+1}, b_{k+1}, \ldots, b_{k+e}\}$ are aligned with a_{m_1}. In other words, for the e frames following the minimum point k, (4) becomes

$$S'(m_1,k+j) = d(m_1,k+j) + S'(m_1,k+j-1) \qquad (10)$$

where $j = 1, 2, \ldots, e$. Then for the true distance S_{true} and the estimated distance $S_{estimated}$, the relative error vector is

$$error = |S_{estimated} - S_{true}| ./ S_{true} \qquad (11)$$

An error tolerance threshold δ is used to measure the relative error. In the experiments we use $e = 20$ and $\delta = 5\%$. If the average relative error Mean($error$) < δ, it is concluded that the local minimum point at k is the global minimum point and therefore the endpoint of this gesture. Otherwise, we continue to test the next local minimum point.

Using this approach, gesture segmentation is implemented in the process of DTW and scores for different gestures can be provided to the user. For each gesture, the extra complexity to test local minimum points is $\Theta(m_1 e)$. Moreover, if $B_1 = \{b_1, b_2, \ldots, b_{n_1}\}$ is determined as the gesture related to the avatar instructor's gesture $A_1 = \{a_1, a_2, \ldots, a_{m_1}\}$, DTW can be conducted from the new starting point $(m_1 + 1, n_1 + 1)$. Figure 5 shows the example of applying GB-DTW on the same sequences as Figure 4. Suppose that there are four gestures in the exercise, segmentation allows DTW to be performed separately for each gesture. The shaded area shows the computation cost for each gesture.

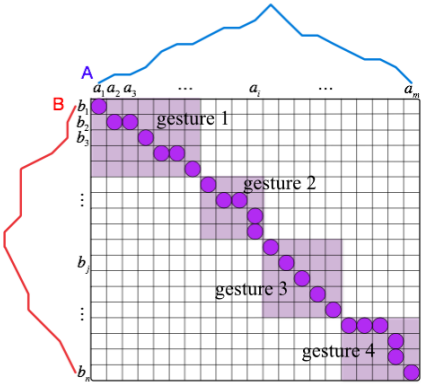


Figure 5. Computation complexity of GB-DTW in an exercise of four gestures

For some training exercises, the motion sequences of the user and avatar instructor might be quite long and the default DTW requires large computation complexity $\Theta(mn)$. Suppose that there are g gestures in a training exercise, so each gesture of the avatar instructor contains $\Theta(m/g)$ frames and each gesture of the user contains $\Theta(n/g)$ frames. The complexity of DTW on each gesture is $\Theta(mn/g^2)$. Besides, for each gesture we need $\Theta(em/g)$ complexity to test local minimum points. So the total complexity of GB-DTW is

$$\Theta(g \times \frac{mn}{g^2}) + \Theta(g \times \frac{em}{g}) = \Theta(m(\frac{n}{g} + e)) << \Theta(mn) \qquad (12)$$

when g is large, the proposed GB-DTW algorithm can significantly decrease the computation complexity compared to default DTW on the entire sequence.

4.4 Motion Data Rescaling Using GB-DTW

Based on the alignment result given by the optimal warping path in each gesture, we can rescale the two motion sequences nonlinearly on the time axis to match them. When multiple adjacent frames in one sequence are aligned with one single frame in the other sequence, the single frame will be repeated for several times. For example, if $\hat{A} = \{a_i, a_{i+1},\ldots, a_{i+w-1}\}$ of the avatar instructor are aligned with b_j of the user, w-1 frames identical with b_j will be inserted after frame j. In this way, the user's movement in each frame matches the corresponding movement of the avatar instructor.

4.5 Accuracy Threshold and Real-Time Guidance

As mentioned above, the proposed GB-DTW algorithm enables the user to receive score for his performance on each gesture in real time. If the user receives a score below a certain threshold for the previous gesture, the system will pause the training video and provide guidance to help the user calibrate his movements. To determine the optimal threshold for the evaluation score, we conduct an experiment with the assistance of a licensed physical therapist with six years of experience.

4.5.1 Accuracy Threshold Determination

In the experiment 10 subjects (aged 18~30, 7 males, 3 females) are required to perform a gesture designed by the physical therapist for nine times. For one performance of each subject, he receives an evaluation $Y \in \{0,1\}$ from the physical therapist where $Y = 0$ represents good performance and $Y = 1$ indicates that he fails the gesture. In the meantime, the proposed cloud-based virtual training system captures the subject's movement, processes the motion data and provides an evaluation score $S \in [0,100]$. Therefore we have a positive dataset $\{S \mid Y = 1\}$ and a negative dataset $\{S \mid Y = 0\}$. According to Bayesian Decision Theory [12], the optimal classification threshold for the two classes is

$$P_{S|Y}(s \mid 0)P_Y(0) = P_{S|Y}(s \mid 1)P_Y(1) \qquad (13)$$

where $P_Y(y)$ is the prior probability of each class. Assuming that the two classes are Gaussian-distributed,

$$P_{S|Y}(s \mid y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp[-\frac{(s-u_y)^2}{2\sigma_y^2}] \qquad (14)$$

where μ_y is the sample mean and σ_y^2 is the sample variance of class y. Form (13) and (14) we get

$$\frac{(s-\mu_0)^2}{\sigma_0^2} - \frac{(s-\mu_1)^2}{\sigma_1^2} + \log(2\pi\frac{\sigma_0^2}{\sigma_1^2}) - 2\log\frac{P_Y(0)}{P_Y(1)} = 0 \qquad (15)$$

The solution s_0 of (15) is the optimal threshold for the evaluation score S given by the system. From the experiment, we get s_0 =

62.8 thus scores below 62.8 demand real-time guidance from the system.

4.5.2 *Visual and Textual Guidance*

In this section, we will discuss how we use the results of GB-DTW to provide visual and textual guidance to the user.

First, we will discuss different alignment types in the result of GB-DTW. Here we define the monotonicity of a subsequence $\hat{A} = \{a_i, a_{i+1}, \ldots, a_{i+w-1}\}$ as follows. If all the features of $\hat{A}$ are monotonic (i.e. keep increasing or decreasing) then $\hat{A}$ is monotonic, or else it is non-monotonic. Suppose that all the frames in $\hat{A} = \{a_i, a_{i+1}, \ldots, a_{i+w-1}\}$ are aligned to b_j, then there are two different cases. (a) If $\hat{A}$ is monotonic, it means that the effects of multiple frames in $\hat{A}$ are similar to the effect of b_j, which indicates that B is faster than A at that time. (b) If $\hat{A}$ is non-monotonic, it means that some reciprocating movements in $\hat{A}$ are aligned to one single frame b_j. Thus, B's gesture is incomplete for this reciprocating motion. Based on different alignment ways between the avatar instructor and the user, we summarize four types of alignments and their corresponding feedback (used as textual guidance for the user): too fast, too slow, overdone gesture, and incomplete gesture. For the arm elevation exercise in our experiments (see Figure 2a), overdone/incomplete gesture means that the user's arm is too high/low. Table 1 and Figure 6 illustrates the four types. For example, in type 1 the user performs faster than the avatar instructor so monotonic subsequence $\{a_3, a_4\}$ of the avatar instructor is aligned with one single frame b_4 of the user. In type 4 the user's gesture does not reach the required amplitude, so non-monotonic subsequence $\{a_{17}, a_{18}, a_{19}\}$ of the avatar instructor is aligned with one single frame b_{21} of the user.

Table 1. Four types of alignment and textual guidance

Type	Number of frames		Monotonicity	Textual Guidance
	Avatar Instructor	User		
1	>1	1	Monotonic	Too Fast
2	1	>1		Too Slow
3	1	>1	Non-Monotonic	Arm Too High
4	>1	1		Arm Too Low

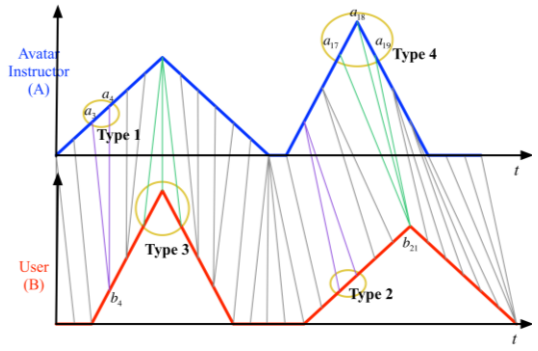


Figure 6. Four types: 1) The user moves faster. 2) The user moves more slowly. 3) Overdone gesture (too high). 4) Incomplete gesture (too low).

Next, we discuss how to calculate accurate evaluation score for each gesture based on the different kinds of training exercises and the types of alignments discussed above. In Section 4.2, $S(m_1, n_1)$ is used to provide evaluation score for the user. However, when the user performs faster or slower than the avatar instructor as type 1 and 2, the difference between the two sequences is counted several times. For example, if all the frames in $\hat{A} = \{a_i, a_{i+1}, \ldots, a_{i+w-1}\}$ are aligned to b_j, then the accumulative distance for this part is

$$\hat{D} = \sum_{k=0}^{w-1} \sqrt{\left| a_{i+k} - b_j \right|^2} \tag{16}$$

However, for some training exercises where speed is not important, the distance should be counted for only once, and (16) can be revised as

$$\hat{D}' = \sqrt{\left| (\frac{1}{w} \sum_{k=0}^{w-1} a_{i+k}) - b_j \right|^2} \tag{17}$$

Therefore, for exercises in which speed is not important, we use (17) to calculate the evaluation score. For exercises where speed should be considered, the original accumulative distance in (16) is used.

After completing one gesture, the user can see the score of his performance on the screen. To better help the user calibrate this performance for any low-score gesture, we propose a replay system to provide two kinds of guidance (visual and textual guidance) for the user. Firstly, the rescaled movements of the avatar instructor together with the rescaled movements of important body parts of the user are shown on the screen. In this way, the user can see the difference of his movements and the avatar instructor's and know how to correct his performance. Secondly, according to the four types in Table 1, textual guidance will be shown on the screen to remind the user about his error type if he made mistakes in speed or movement range of the gesture. (For those exercises in which speed is not important, type 1 and 2 will be ignored.) Figure 10 provides snap shots of some visual and textual guidance provided by our system, and will be explained in the next section.

5. EXPERIMENTAL RESULTS

Our experiments are based on a testbed (shown in Figure 7) we developed to emulate the system architecture in Figure 1. The cloud server is a quad core 3.1GHz CPU with 8GB RAM, and the mobile device is a laptop PC with a dual core 2.5GHz CPU and 4GB RAM. The network connection between the server and the mobile laptop is emulated using a network emulator (Linktropy), which can be programmed to emulate different wireless network profiles.

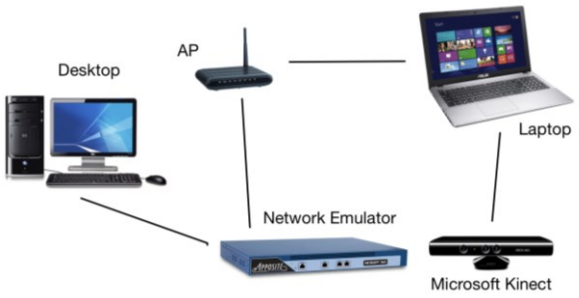


Figure 7. Experimental testbed.

The tested exercise is laterally moving one's left arm from the solid position to the dotted position and then returning to the solid position (shown in Figure 2(a)) with different angle θ for five times. The angle of the left shoulder is measured and five gestures are defined for this exercise. The avatar instructor's motion data for the five gestures are shown as the blue curve in Figure 8.

Four users (User A, B, C and D) with different human reaction delays are invited as subjects in the experiment. They try to follow the avatar instructor by watching the motion video which is transmitted through the network emulator to the laptop. Each user is tested under ideal network condition (without any bandwidth constraint) and non-ideal network condition by using a bandwidth profile. The bandwidth profile is shown as the black solid curve in Figure 8 and it is repeated for each user using the network emulator. It can be seen that the bandwidth is relatively lower at the third and fourth gesture.

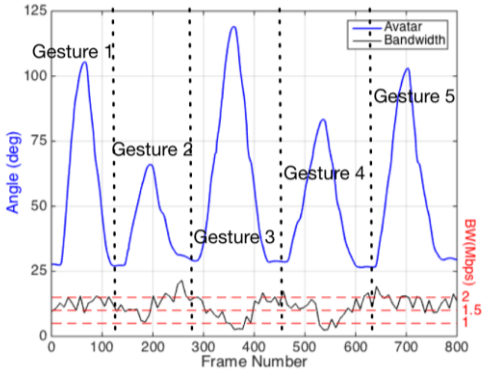


Figure 8. Avatar instructor's motion data and the bandwidth profile.

Then we use three different methods: 1) tradition method of MCC, 2) default DTW on the entire sequence, and 3) GB-DTW, to align the motion sequences of the avatar instructor and the user. The alignment results of user A are shown in Figure 9. In each figure, we plot the motion data of the avatar instructor 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 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 for several times. Therefore, the rescaled sequences are longer than the original sequences. From Figure 9 we can see that User A performs worse with fluctuating bandwidth than ideal network condition due to the network delay. Especially at the third and fourth gesture when bandwidth is limited, he performs more slowly than the avatar instructor. To quantify the alignment results, we calculate the correlation coefficient ρ of the aligned sequences x and y in each method

$$\rho = \frac{E[(x-\bar{x})(y-\bar{y})]}{\sqrt{\sigma_x^2 \sigma_y^2}} \qquad (18)$$

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 2. It can be inferred that the human reaction delay of User A and B is smaller than that of User C and D under ideal network condition. Comparing the three methods, we can conclude that under ideal network condition with only human reaction delay, the traditional method of MCC gives high correlation coefficients (ρ > 0.85). However, when the network condition is not ideal and therefore large network delay is accumulated, the two DTW algorithms perform much better (ρ > 0.95) than MCC (ρ < 0.80). For the default DTW and GB-DTW, their alignment results are quite close and both of their correlation coefficients are more than 0.95. Therefore, the proposed GB-DTW can give perfect alignment results as well as save the computation complexity and enable real-time visual guidance.

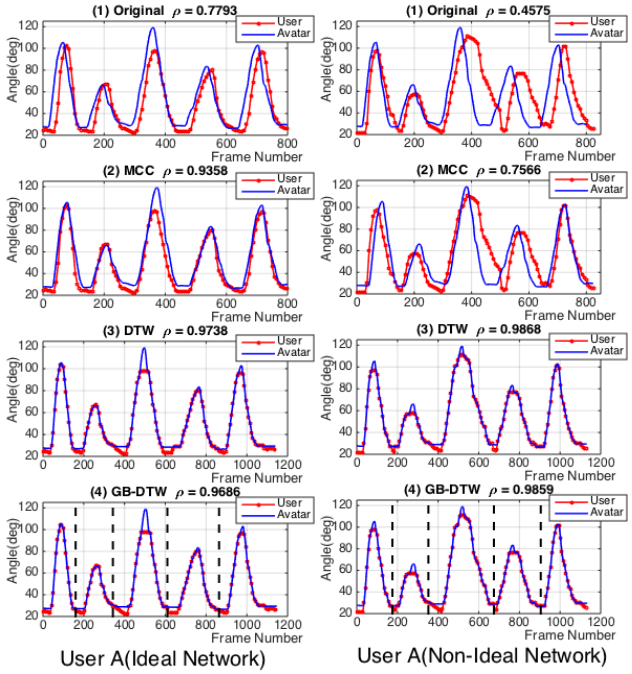


Figure 9. Data alignment results for User A under ideal and non-ideal network condition. (1) Original misaligned motion sequences of the avatar instructor and the user. (2) Shifted sequences using MCC. (3) Rescaled sequences using default DTW on the entire sequence. (4) Rescaled sequences using GB-DTW and the segmentation result.

Table 2. Correlation Coefficients for Users A, B, C, and D using different methods under ideal and non-ideal network condition.

User	Condition	Original	MCC	DTW	GB-DTW
User A	Ideal	0.7793	0.9358	0.9738	0.9686
	Non-Ideal	0.4575	0.7566	0.9868	0.9859
User B	Ideal	0.7824	0.9578	0.9741	0.9741
	Non-Ideal	0.4726	0.6104	0.9811	0.9808
User C	Ideal	0.6388	0.8766	0.9654	0.9575
	Non-Ideal	0.1036	0.6351	0.9888	0.9887
User D	Ideal	0.6190	0.9302	0.9752	0.9674
	Non-Ideal	-0.0944	0.7115	0.9851	0.9816

Apart from the alignment results, Figure 10 shows two examples of textual and visual guidance in this training exercise. Six adjacent frames are provided in each example. The red arm shows the movements of the user. Textual feedback is provided on the bottom of the screen. In Figure 10(a), the user moves his arm higher than required. In Figure 10(b), the user moves more slowly than the avatar instructor when laying down his arm.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a cloud-based virtual training system that captures and evaluates the user's performance automatically.

It can be applied to various training applications, including physical therapy, wellness and fitness training, ergonomics training, etc. To address the data misalignment problem caused by human reaction delay and network delay, we propose the GB-DTW algorithm to align the two sequences as well as segment sequences into gestures in real time. Experiments with multiple subjects under real network condition show that the proposed method works better than other evaluation methods and can provide detailed visual and textual guidance for the user. However, the proposed method may fail to provide exact segmentation and evaluation results when the user is not following the avatar instructor at all. In future work, we will try to solve this problem and improve the proposed segmentation approach. Besides, we will upgrade the system to support more kinds of useful guidance and more real-time evaluation (react right after each frame) and include more subjects to validate the results.

7. REFERENCES

[1] Kinect: www.xbox.com/en-US/kinect

[2] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz, "Using wearable sensors for real-time recognition tasks in games of martial arts-an initial experiment," *Computational Intelligence and Games* (CIG'06), Reno, May, 2006.

[3] D. Jack, et al., "Virtual reality-enhanced stroke rehabilitation," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 9.3 (2001): 308-318.

[4] A. Mirelman, B. L. Patritti, 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.

[5] C. Y. Chang, et al., "Towards pervasive physical rehabilitation using Microsoft Kinect," *Pervasive Computing Technologies for Healthcare* (PervasiveHealth'12), San Diego, May, 2012.

[6] 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.

[7] 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.

[8] A. Shapiro, et al., "Rapid avatar capture and simulation using commodity depth sensors," *Computer Animation and Virtual Worlds*, 25.3-4 (2014): 201-211.

[9] D. J. Berndt, and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," *KDD workshop*, Vol. 10. No. 16. 1994.

[10] M. Müller, "Information retrieval for music and motion," Vol. 2. Heidelberg: Springer, 2007.

[11] 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.

[12] James O Berger, "Statistical decision theory and Bayesian analysis," *Springer Science & Business Media*, 1985.

[13] M. T. Nkosi, and F. Mekuria, "Cloud computing for enhanced mobile health applications," *Cloud Computing Technology and Science* (CloudCom'10), Indianapolis, December, 2010.

[14] R. Andre-Obrecht, "A new statistical approach for the automatic segmentation of continuous speech signals," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36.1 (1988): 29-40.

[15] P. Fearnhead, "Exact Bayesian curve fitting and signal segmentation," *Signal Processing, IEEE Transactions on,* 53.6 (2005): 2160-2166.

[16] S. Wang, and S. Dey. "Cloud mobile gaming: modeling and measuring user experience in mobile wireless networks," *ACM SIGMOBILE Mobile Computing and Communications Review* 16.1 (2012): 10-21.

[17] The U.S. Mobile App Report by comScore: https://www.comscore.com/Insights/Presentations-and-Whitepapers/2014/The-US-Mobile-App-Report

Figure 10. Examples of textual and visual guidance. The rendered red arm shows the user's left arm movement relative to the avatar instructor's, after motion data alignment using proposed GB-DTW technique. Textual guidance is provided on the bottom. (a) The user's arm is higher than required. (b) The user moves more slowly than the avatar instructor when moving his arm down.