IEEE Journal of Translational Engineering in Health and Medicine

Received 12 March 2018; revised 12 June 2018; accepted 13 July 2018. Date of publication 12 October 2018; date of current version 12 November 2018.

Digital Object Identifier 10.1109/JTEHM.2018.2859992

Kinect-Based In-Home Exercise System for Lymphatic Health and Lymphedema Intervention

AN-TI CHIANG[1], QI CHEN[1], YAO WANG[1], AND MEI R. FU[2]

[1]Department of Electrical and Computer Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201, USA
[2]NYU Rory Meyers College of Nursing, New York, NY 10010, USA

CORRESPONDING AUTHOR: A.-T. CHIANG (atc327@nyu.edu)

This work was supported by the National Institutes of Health under Grant R01CA214085.

ABSTRACT Using Kinect sensors to monitor and provide feedback to patients performing intervention or rehabilitation exercises is an upcoming trend in healthcare. However, the joint positions measured by the Kinect sensor are often unreliable, especially for joints that are occluded by other parts of the body. Also, users' motion sequences differ significantly even when doing the same exercise and are not temporally aligned, making the evaluation of the correctness of their movement challenging. This paper aims to develop a Kinect-based intervention system, which can guide the users to perform the exercises more effectively. To circumvent the unreliability of the Kinect measurements, we developed a denoising algorithm using a Gaussian Process regression model. We simultaneously capture the joint positions using both a Kinect sensor and a motion capture (MOCAP) system during a training stage and train a Gaussian process regression model to map the noisy Kinect measurements to the more accurate MOCAP measurements. For the sequences alignment issue, we develop a gradient-weighted dynamic time warping approach that can automatically recognize the endpoints of different subsequences from the original user's motion sequence, and furthermore temporally align the subsequences from multiple actors. During a live exercise session, the system applies the same alignment algorithm to a live-captured Kinect sequence to divide it into subsequences, and furthermore compare each subsequence with its corresponding reference subsequence, and generates feedback to the user based on the comparison results. Our results show that the denoised Kinect measurements by the proposed denoising algorithm are more accurate than several benchmark methods and the proposed temporal alignment approach can precisely detect the end of each subsequence in an exercise with very small amount of delay. These methods have been integrated into a prototype system for guiding patients with risks for breast-cancer related lymphedema to perform a set of lymphatic exercises. The system can provide relevant feedback to the patient performing an exercise in real time.

INDEX TERMS Dynamic time warping, Intervention system, denoising of Kinect measurements, Gaussian process regression.

I. INTRODUCTION

Having patients performing prescribed exercises is an important clinical intervention for many health conditions such as chronic pain management, post-surgery rehab, and physical therapy after a sports injury. Using sensor-based systems to automatically track patients' movements during their exercises and to provide instant feedback to the patients regarding the ‘‘correctness’’ of their movements holds great promise in reducing the cost for such interventions and increasing their effectiveness.

Generally, a gold standard for motion sensing is the so-called Motion Capture (MOCAP) system. It consists of multiple cameras positioned in a specially designed room to capture the 3D positions of reflective markers put on an actor's clothing. Such a system can track human action accurately but it is not a practical solution for clinical use since users have to wear tight suits with markers and such systems are very expensive [1].

Compared with the expensive motion capture system, the Kinect sensor by Microsoft [2] (or similar sensors such as Realsense by Intel [3]) is more affordable and convenient to most of the clinics or even in patients' homes. Kinect contains an RGB camera, an infrared sensor and infrared emitters, which can be used to estimate the depth map of the imaged scene. Kinect further provides an SDK that can estimate the joint positions in the 3D domain. However, because Kinect derives the 3D positions from a single view point, the estimated positions of some joints are not accurate, when these joints are occluded or partially occluded.

A number of works have been reported for stabilizing joint tracking and improving pose reconstruction. Shum *et al.* proposed an optimized data driven method to solve posture reconstruction problem [4]. Here a posture is defined by the 3D positions of all joints at any time. They first collect user postures while the users are performing an exercise during a training stage. Second, they remove similar postures in the training dataset by thresholding the sum of squared differences of each posture sample pair. Then they apply Principle Component Analysis (PCA) on remaining postures to reduce dimensions. Once Kinect captures a noisy posture, the system first projects the captured postures to the PCA bases, and then reconstructs a denoised posture from the projection coefficients and their corresponding PCA bases. The reconstruction results heavily depend on the training database. Wei *et al.* formulated the denoising problem into a Maximum A Posteriori (MAP) estimation problem based on the Kinect depth image [5]. Although the proposed algorithm can be implemented on a GPU in multithreading to accelerate the speed, it needs to initialize the starting pose manually and it sometimes will be stuck at the local minimum, which makes reconstruction fail. Liu *et al.* on the other hand, proposed to use Gaussian process regression to reconstruct the body joints [6]. They generate several Gaussian process models, one for each joint and each model is generated by only considering the joint and its nearest neighborhood joints. Such an approach cannot exploit the typical relations among all body joints during a particular exercise. Also, the distance between two connected joints in the denoised results is not guaranteed to follow the bone length constraint. Gaussian process regression has also been employed for image super-resolution successfully [7]. Tripathy *et al.* proposed to use Kalman filter with a bone length constraint between every two connected joints [8]. This algorithm can make the joints trajectories smoother and preserve the joints' kinematic characteristics. However, because the algorithm considers one segment at a time, it does not effectively exploit the relationship among multiple joints during a human action.

To evaluate the performance of the user's motion with a reference motion derived from a training dataset from the database, Alexiadis *et al.* [9] proposed to apply the maximum cross correlation to find the offset between user's motion sequence and a ground truth sequence. Then, by applying this offset to the user's motion data, the two sequences are aligned and their similarity can be calculated. However, this method applies one shift to the whole sequence and can not deal with the situation where users may have inconsistent speed when performing different parts of an exercise. Yurtman *et al.* [10] apply the dynamic time warping to detect and identify accurate and inaccurate implementations of a physical therapy exercise. However, this system requires the user to attach wearable motion sensors, which is very expensive and inconvenient for the user. Jeong *et al.* [11] proposed a modified penalty-based dynamic time warping algorithm, which considers the phase difference between the reference sequence and the test sequence. Although this method is robust to outliers, its computation cost is very high, as does the original DTW method, which makes it unfeasible for real time applications. Su *et al.* [12] proposed a Kinect-based rehabilitation system by using the DTW and fuzzy logic. The rehabilitation system can use the DTW algorithm to compare the ''in-home'' and ''in clinic'' sequence. The system needs to wait for the user to finish an entire exercise before starting the evaluation, and hence is not capable of instantaneous feedback. Wei *et al.* [13] consider a cloud based system, where a user can download the exercise video from the cloud database to the local device. Then, the user employs a Kinect sensor to capture his/her motion data and upload the data back to the cloud for a remote server to evaluate the correctness of the exercise and provide feedback. They proposed a gesture-based DTW algorithm to align the patient motion with a reference motion. In their work, the exercise is relatively easy and only focus on one joint, so the remote motion analysis algorithm only needs to consider one joint. Our system is developed to handle complicated exercises where multiple joints need to be analyzed, both for temporal alignment and for assessment of motion correctness.

The rest of the paper is organized as follows. In Sec. II, we describe the proposed system. First, we briefly describe how do we collect and preprocess our data. Then we present the proposed clustered Gaussian process regression method for motion sequence denoising. Next, we discuss the structure of the lymphatic exercises and explain how to generate the reference subsequence. Finally, we discuss our low-delay DTW algorithm for temporal alignment of the motion sequence with a reference sequence. Section III presents the overall system design, and presents the experimental results and the overall system design. We conclude this paper in Sec. IV. A preliminary version of Sec. II-A and Sec. II-B has been reported in [1].

II. THE PROPOSED SYSTEM AND METHODS

In this work, in order to generate reference data both for denoising of Kinect data, and for evaluating the correctness of user motion, during a training stage, we first capture the joint movement traces of different people performing the same exercise by using a motion capture system and a Kinect V2 sensor simultaneously. Second, we convert each captured trace to a standardized domain to eliminate the bias due to

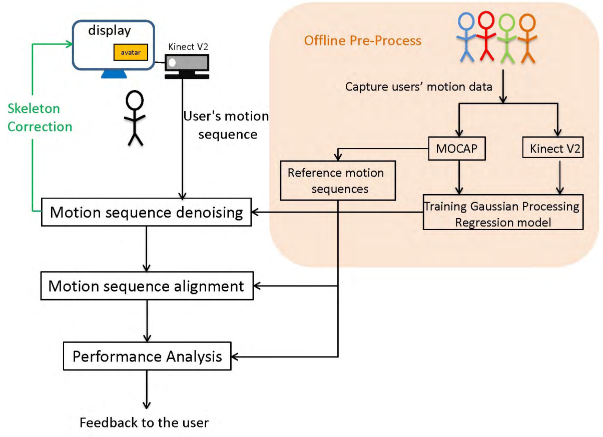
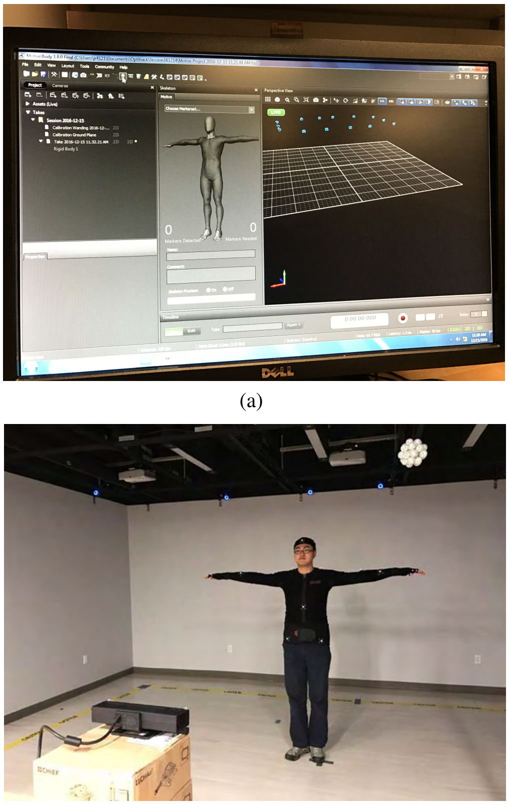

display
avatar
Kinect V2
Skeleton Correction
User's motion sequence
Offline Pre-Process
Capture users' motion data
MOCAP
Kinect V2
Reference motion sequences
Training Gaussian Processing Regression model
Motion sequence denoising
Motion sequence alignment
Performance Analysis
Feedback to the user

FIGURE 1. Proposed system flowchart.

the different body sizes of the users. Using these standardized data, we train a Gaussian process regression model using both the standardized MOCAP data and the standardized Kinect data captured during the training stage. This model then can be used to predict the unavailable MOCAP measurement (consisting of 3D positions of all joints of interests) from the Kinect measurement captured during an actual exercise session.

To enable the comparison of a user's motion sequence (defined as the sequence of the 3D positions of target joints) during an exercise with some "gold standard", we recognize that each exercise can be divided into multiple subsequences each corresponding to a portion of the exercise transitioning from one pose to another. We develop a dynamic time warping approach that can automatically recognize the endpoints of different subsequences, and furthermore temporally align the subsequences from multiple actors. To generate a reference motion sequence for each exercise, an expert performed each exercise several times during the training stage. The recorded MOCAP sequences are divided into subsequences and the corresponding subsequences are temporally aligned and then averaged to yield the reference subsequence. During a live exercise session, the system applies the same alignment algorithm to a live-captured Kinect sequence to divide it into subsequences, and furthermore compare each subsequence with its corresponding reference subsequence, and generates feedback to the user based on the comparison results. The developed alignment algorithm allows us to detect whether a user is performing each exercise correctly, while taking into account the possible speed mismatch between the user and the expert.

The proposed intervention system has two major parts, as shown in Fig. 1. The first part is the training stage, during which we use both a Kinect sensor and a motion capture (MOCAP) system to capture the joint positions simultaneously and train a Gaussian Process regression model to map the noisy Kinect measurements to the more accurate MOCAP measurements. Also, we use the MOCAP data recorded from an expert to generate the reference sequence for each exercise.


(a)

(b)

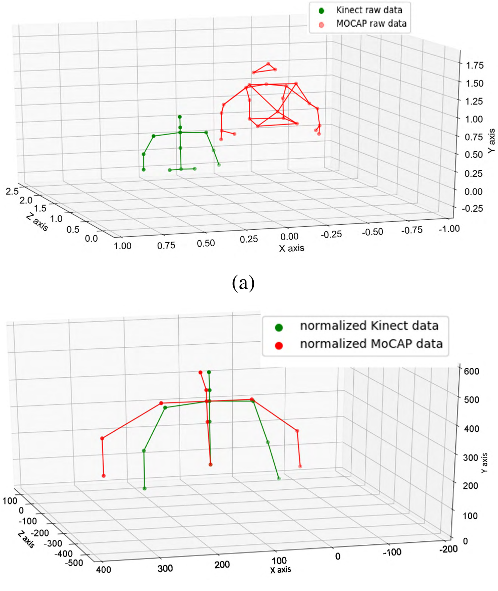
FIGURE 2. Data capture system set up [1]. (a) software interface of MOCAP system, (b) a typical recording scenario.

The second part is for the live session when a patient (to be referred to as a user) is performing the intervention exercises and the Kinect sensor captures both an RGB video of the user and the joint motion. The system has a display screen (see Fig. 1) that shows an avatar performing a target exercise and the live captured video of the user with an overlay of the skeleton connecting the denoised joint positions. The system will compare the captured motion sequence with the reference sequence established during the training stage, and provide instantaneous feedback to the user regarding the improvement needed after processing each subsequence. The system will also provide constructive feedback at the end of an exercise. We have developed a prototype system for the second part, which can operate in real time while a user is performing an exercise.

A. MOTION DATA STANDARDIZATION AND DENOISING

1) DATA CAPTURE AND STANDARDIZATION

We employ a motion capture system called "Motive" with 14 cameras [14] to record 25 marker positions on the upper body of a volunteer as shown in Fig. 2(b). At the same time, we place a Kinect V2 sensor in the front of the volunteer to capture the color image (resolution 1920x1080), depth image (resolution 512x424) and joint positions simultaneously. The capture system is shown in Fig. 2. The number of joints captured by MOCAP system differs from the joints



(a)

(b)

FIGURE 3. **Upper body skeletons [1]. (a) raw MOCAP sample and raw Kinect sample. (b) standardized MOCAP sample and Kinect sample.**

captured by Kinect sensor. To handle the problem, we convert the MOCAP joint positions to Kinect-like joint positions as described in [1] by referring the human Anthropometry feature [15]. To circumvent the wide variations in the limb lengths between different users, we first convert all Kinect and MOCAP data into a standardized domain so that the distance between every two connected joints is fixed at some predefined length as shown in Fig. 3. More details about the data standardization can be found in [1]. Note that the joint position denoising and the temporal alignment and comparison with the reference sequence (Sec. II-C) are accomplished in the standardized domain.

2) TRAINING AND TESTING DATA SETS

The The-Optimal-Lymph-Flow^{TM} (TOLF) exercise is developed by Dr. Mei R. Fu and her research team. TOLF exercise is a patient-centered exercise program focusing on lymphatic health by promoting lymph flow [16]–[18]. Appendix A provides details for the TOLF exercise. Currently, we only focused on a subset of the TOLF exercises that require the tracking of the upper body joints, which includes spine shoulder, left and right shoulders, left and right elbows and left and right wrists.

It should be noted that large muscle exercises such as walking, dancing, Yoga, are also part of TOLF exercise program but not part of current Kinect-based system due to the complexities of the exercises. Yet, this Kinect system is an initial step towards using Kinect or other similar sensors to deliver and monitor large muscle exercises such as Yoga or specific dance.

TABLE 1. **Training and testing data set [1].**

Name of the exercise	Number of training samples	Number of testing samples	
		Reliable data	unreliable data
muscle-tightening deep breathing (Exercise 1)	22252	5564	3514
over the head pumping (Exercise 2)	22425	5607	2436
push down pumping (Exercise 3)	21664	5416	3222
horizontal pumping (Exercise 4)	22600	5651	4400

A total of 14 healthy volunteers were recruited among students in NYU for motion data collection and their age range are between 23 - 30. Each volunteer performed each lymphatic exercise 3 to 7 times, while being recorded by a Kinect sensor and the ''Motive'' MOCAP system simultaneously. The frame rate of the motion capture was 30 frames/sec in both systems. After removing some corrupted recordings, we have around 70 pairs of motion traces for each exercise, each containing 400-900 samples. Each sample refers to the measured positions of all joints of a user at a single sampling time. These samples are divided into training samples and testing samples for the purpose of training and testing the Gaussian process regression model for denoising. The distributions of samples for different exercises are summarized in Table 1. All healthy volunteers consent to publication of material about them.

3) DENOISING USING GAUSSIAN PROCESS REGRESSION MODEL

Let x_t and y_t represent the Kinect sample and the MOCAP sample at time t separately. To denoise the Kinect data, we train a Gaussian process regression (GPR) model that maps the Kinect sample x_t to the MOCAP sample y_t. A different model is trained for each exercise with all the training data for this exercise, where each training sample is a pair of corresponding x_t and y_t. with all training samples [19].

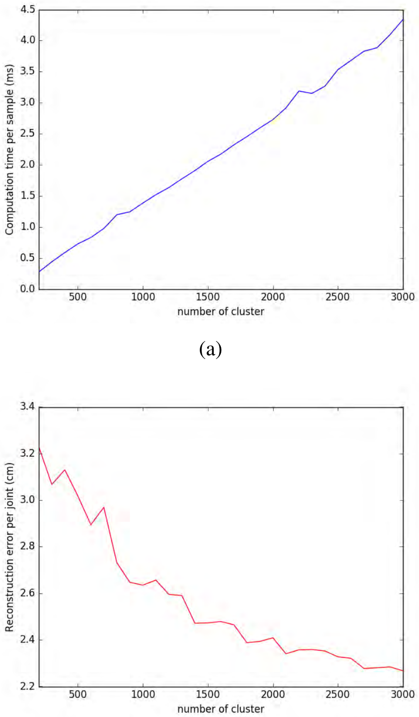
Since the trained Gaussian regression model does not always preserve the limb length between connected joints, we apply the standardization method, described in Sec. II-A.1, to the resulting joint positions, so that the distance between every two connected joints follows the reference length.

4) CLUSTERED GAUSSIAN PROCESSING

Once the GPR model is trained, we have the hyper-parameters of the model. In the denoising stage, we only need to compute between the input sample and each training sample. Therefore, the computation time for denoising is linearly increasing with the number of training samples. Our training data consists of around 22000 samples for each exercise, making the denoising process very slow. In order to reduce the computation time, Cao *et al.* select a subset of training samples following a chosen optimization criterion and form a sparse Gaussian Process model [20]. Snelson *et al.* derive the model with a set of pseudo input and other model parameters [21]. In this paper, we apply classical K-means method to the training samples and separate the original data samples into Q different clusters. Then, we train the GPR model using

TABLE 2. Average reconstruction error per joint by different denoising methods (in unit of cm) [1].

Name of the exercise		Number of samples	Raw Kinect data	Liu et al. [6]	Proposed method
muscle-tightening deep breathing	All testing data	9018	10.22	4.91	1.46
(Exercise 1)	Unreliable testing data	3514	13.82	6.49	3.1
over the head pumping	All testing data	8043	9.21	8.18	1.76
(Exercise 2)	Unreliable testing data	2436	13.12	11.89	6.23
push down pumping	All testing data	8638	8.94	7.36	3.38
(Exercise 3)	Unreliable testing data	3222	13.57	11.54	7.45
horizontal pumping	All testing data	10051	8.38	6.27	2.7
(Exercise 4)	Unreliable testing data	4400	13.79	10.13	6.76



(a)

(b)

FIGURE 4. Computation time and error for different cluster numbers [1]. (a) average computation time per time sample, (b) error per joint.

a reduced training set containing only the cluster centroids. The performance using this approach under different cluster numbers is shown in Fig. 4. As expected, the regression error (for exercise 4 in Table 2) decreases as the cluster number increases. But we found that $Q = 800$ is a sweet spot, obtaining a local minimum in the regression error, and yet having a relatively short computation time. The computation time here is measured when the algorithm runs on a Windows 7 computer with Intel i5-4570 CPU and 16GB RAM.

5) RELIABILITY

In order to train the GPR using only reliable data samples, we define the reliability for each joint position. We consider kinematic reliability, temporal reliability and tracking reliability reported by Kinect, and combine them to evaluate the overall reliability. Only if the reliability score of a joint is larger than the threshold, this joint is considered reliable. In the training stage, we only choose samples with all reliable joints to train the Gaussian regression model. In the testing stage, if a time sample contains any unreliable joint, this sample will be denoised. More details about the definition of joint reliability can be found in [1].

B. EXERCISE SEQUENCE DECOMPOSITION AND REFERENCE SEQUENCE GENERATION

1) DECOMPOSITION OF A MOTION SEQUENCE

Usually, an exercise contains a series of movements. Here, we define one exercise as a time sequence, and each time sample is one static pose, defined by the 3D positions of all 7 joints which include left/right shoulders, left/right elbows, left/right wrists and spine shoulder. There are lots of poses in one exercise, but usually we will focus on certain key poses. The original exercise can be decomposed into several key poses and the transition between two key poses. We define the transition from one key pose to the next key pose as a subsequence. Furthermore, an exercise usually contains several repetitions (REP) of the same set of ordered subsequences. For example, the exercise ‘‘horizontal pumping’’ in our TOLF exercise set as shown in Fig. 10(b), contains four major subsequences, first is from ‘‘hands down’’ to ‘‘T-pose’’, second is from ‘‘T-pose’’ to ‘‘hands close to the chest’’, the third subsequence is from ‘‘hands close to the chest’’ to ‘‘T-pose’’ and the final one is from ‘‘T-pose’’ to ‘‘hands down’’. During the ‘‘horizontal pumping’’ exercise, users do the first subsequence at the beginning. Then, they do the subsequence 2 and subsequence 3 repeatedly four times. This will be followed by subsequence 4, which finishes the whole exercise.

2) REFERENCE SEQUENCE GENERATION FOR EACH EXERCISE

For each exercise, we generate a reference sequence, which consists of multiple subsequences and repetitions, and each repetition further consists of multiple subsequences. As described in Sec.II-A.2, we ask an expert to perform each exercise several times and record the Kinect and MOCAP motion traces for each exercise. There are around 400 - 900 time samples in each trace. We use the MOCAP data to create the reference sequence. The raw MOCAP data in one subsequence are shown as Fig. 5(a). In Fig. 5(a), all subsequences are captured from the same expert, but each subsequence has different length. This is because it is hard to use the same speed every time when one does the exercise. To deal with this problem, we need to normalize the raw

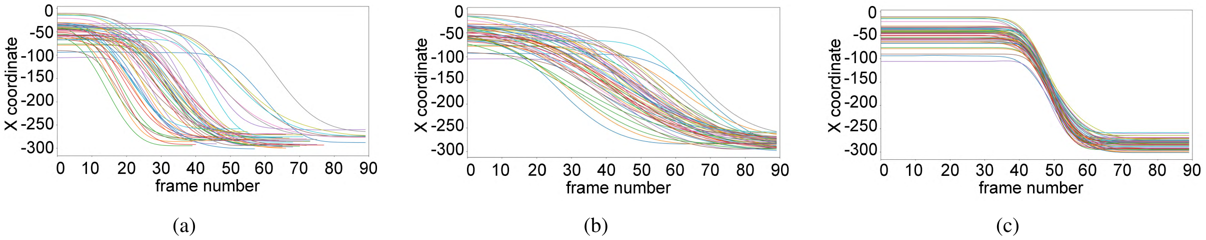


FIGURE 5. Traces of the left wrist x-coordinate while users performing a subsequence in the "Horizontal Pumping" exercise. (a) Standardized MOCAP traces; (b) Aligned traces by stretching all traces to the same length; (c) Aligned traces by aligning at two key transition points.

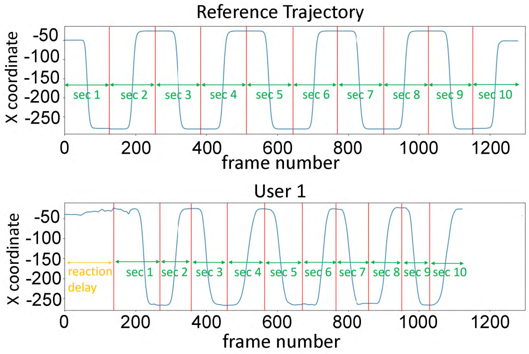


FIGURE 6. The reference sequence and a user's motion sequence of the x coordinate of the left wrist during Exercise 4. Each"sec" indicate a subsequence.

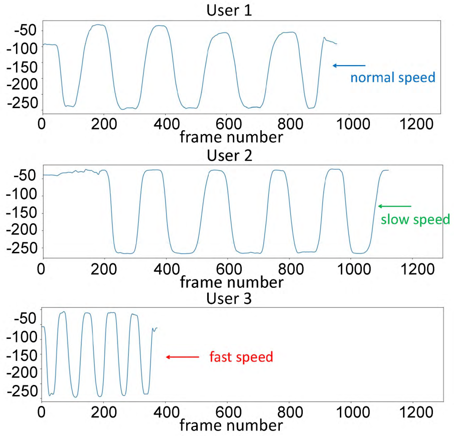


FIGURE 7. Motion data of different users.

MOCAP data to the same length before we average these data to create a single reference. Although we can just interpolate all the raw data to the same length, as shown in Fig. 5(b), they are not aligned in where the transition occurs. To circumvent this problem, we apply second order derivative to the raw data and find two turning points.

According to these two turning points, we divide the subsequence into three parts. We map each part to an assigned length and then combine three parts together to get our normalized subsequences as shown in Fig. 5(c). Finally, we use the mean of the normalized subsequences from multiple expert traces performing the same subsequence to generate the reference sequence for this subsequence.

3) EVALUATION OF A USER'S MOVEMENT AGAINST THE REFERENCE SEQUENCE

For each exercise, we predefine a series of subsequences (see the example for the "horizontal pumping" exercise given in Sec. II-B.1). When a user performs an exercise, her joint motion sequence will be captured by the Kinect sensor. Our system will denoise the user sequence and compare the denoised sequence with the reference subsequences sequentially, to determine the endpoint of each subsequence. The system will further analyze the difference between each identified user subsequence with the corresponding reference subsequence, to determine what feedback to provide to the user. More details will be discussed in Sec.II-C.

C. TEMPORAL ALIGNMENT USING LOW-DELAY DYNAMIC TIME WARPING

1) HUMAN REACTION DELAY AND MOTION VARIABILITY

In our system, a user is supposed to follow the movement of the avatar shown on the display during each exercise. Each exercise typically consists of multiple repeats of the same movement. Usually, at the beginning of an exercise, the user may take a few seconds to understand what should she do before starting to follow the avatar's movement. Fig. 6 shows the reference sequence and a user's motion sequence during the exercise. We can see that the user takes some reaction time to figure out what kind of motion she needs to follow in the beginning. After the user has learnt what to do in each repeat, the user tends to spend less reaction time to do the following subsequences. Also, once a user learns what to do, she may do each repeat faster or slower than the avatar. From Fig. 7 we can see that different users will perform the same exercise or each subsequence with different speeds. We will discuss how to deal with these problems in the following subsections.

2) DYNAMIC TIME WARPING FOR MOTION SEQUENCES

Dynamic time warping (DTW) is a popular algorithm for measuring the similarity between two temporal sequences, which may vary in their temporal dynamics. DTW are widely used in temporal sequence matching [22]–[27]. DTW measures the similarity between two given sequences by finding the optimal correspondence between sampling points in the two sequences with certain restrictions. The original DTW method was developed for aligning two sequences of scalar variables (e.g. audio signal intensity). Here, we extend it to align two sequences of vector variables $A = [a_1, a_2, \ldots, a_M]$, and $B = [b_1, b_2, \ldots, b_N]$, where a_i and b_i each represents the 3D positions of 7 joints at time sample i. We define a $M \times N$ distance matrix with the (m, n)-th entry being the Euclidean distance between a_m and b_n is, i.e., $d(m, n) = ||a_m - b_n||_2$. To find the best way to map sequence A to sequence B, a continuous warping path is found by minimizing the summation of the distance on the path. The final DTW path is defined as $P = [p_1, p_2, \ldots, p_q]$, where $max(M, N) \leq q \leq M + N - 1$ and $p_k = (m_k, n_k)$ indicates that a_{m_k} is mapped to b_{n_k} in the path. The optimal DTW distance is

$$S(M, N) = \sum_{(i,j)\in path} d(i, j) \tag{1}$$

Directly using the DTW on two sequences to evaluate their similarity can be affected by the absolute difference in the data amplitude of the two sequences. Following [13], given two sequences $A = [a_1, a_2, \ldots, a_M]$ and $B = [b_1, b_2, \ldots, b_N]$, we first find the difference c between the two initial elements in sequence A and B, where $c = a_1 - b_1$. We then generate the normalized sequence B' with $B' = B + c$. We will apply DTW to A and B' to find the optimal correspondence path, and use the resulting path to align A and B.

3) DTW FOR SUB-SEQUENCE DETECTION AND ALIGNMENT

As introduced in Sec. II-B.1, each exercise can be divided into multiple subsequences. It is better to give some feedback after a user has just finished each subsequence, rather than after the user has finished the entire exercise. Therefore, we need to develop an algorithm that can automatically detect the end of each subsequence soon after it is done, and furthermore align this subsequence with its corresponding reference subsequence. We accomplish this goal by modifying the original DTW to a subsequence-based DTW. Assume that the reference motion sequence for this exercise contains K subsequences and is denoted as $A = [A_1, A_2, \ldots, A_K]$, with $A_k = [a_{k,1}, a_{k,2}, \ldots, a_{k,M_k}]$ and the user's motion sequence for this exercise is $B = [b_1, b_2, \ldots, b_N]$. To determine the endpoint, q, of the first subsequence of the user, we compute the DTW distance between each candidate subsequence $[b_1, b_2, \ldots, b_q](q = 2, 3 \ldots, N)$ with the first subsequence $A_1 = [a_{1,1}, \ldots, a_{1,M_1}]$ of the reference sequence and find q that minimizes this distance, i.e.,

$$N_1 = \arg\min_{q} \; S(M_1, q) \tag{2}$$

Directly solving (2) means that the endpoint of the first subsequence cannot be decided until we go through the entire user sequence, which is very time consuming and prevent us from giving instantaneous feedback to the user. To overcome this problem, we propose a sequential decision approach. Let the initial time when the recording of an exercise starts be 0. At time 0, the user's sequence is $B = [b_0]$ and in sample time 1 the user's sequence is $B = [b_0, b_1]$. When the length of B is larger than 2, we calculate the DTW distance between A_1 and the user's current sequence. We keep capturing the user sequence until a certain time $t^* + 1$ when we find the DTW distance $S(M_1, t^*)$ between A_1 and sequence $B = [b_0, b_1, \ldots, b_{t^*}]$ reaches a minimum value at t^*. That is $S(M_1, t^*) < S(M_1, t^* + 1)$ and $S(M_1, t^*) < S(M_1, t^* - 1)$. To ensure the current point t^* is not a poor local minimum due to noise, we will keep comparing the DTW distance in the following T frames(in this paper we set T as 15). If there is no DTW distance less than the DTW distance at time t^*, then we set time t^* as an endpoint. Otherwise, we will keep looking for the local minimum in the following time points beyond the T frames. Then we reset the current time to 0, and starts to compare the second subsequence A_2 of the reference sequence with the new samples of the user's sequence, to identify the endpoint of the second subsequence.

Ideally, the user should hold her pose in the starting and ending point of each subsequence. However, sometimes the user may slightly move the body during this pause period. This "pause-move-pause" action may make the system to falsely assume that the user has already finished the current subsequence, and starts to look for the endpoint for the next subsequence. To deal with this problem, the system will not start a new DTW process for the next subsequence unless the user's motion is larger than a threshold. The user's motion is measured by the sum of the weighted joints pixel difference(we set the threshold as 2000 in our current system).

4) SPEED UP OF DTW

The DTW algorithm's complexity for subsequence k is $O(M_k N_k)$, which is proportional to the length of the user's subsequence N_k. Therefore, the slower the user's motion, the more computation time the system will spend. To deal with this problem and to accelerate the algorithm, we first apply the DTW algorithm every L frames. That is we downsample both the reference subsequence and the user sequence by a factor of L. In our work, we set L as 10. After the system finds the initial endpoint at time t^* following the approach described in Sec. II-C.3 We will further check the time points between $t^* - L$ to t^* in the original sequences to find the best endpoint. Instead of using the DTW distance to decide on the optimal endpoint, we look for a mid point in the pause period, which should have the least amount of joint motion. We measure the joint motion by the weighted sum of the temporal gradient magnitudes of the joint positions. That is, we find the end time point using

$$t^{\dagger} = \arg\min_{t\in(t^*L,t^*)} \sum_{i\in joints} w_{exercise_n}(i) * ||b_{t,i} - b_{t-1,i}||_2 \tag{3}$$

We assign different weights to different joints based on the characteristics of each exercise. We set the weighted value of highly related joint as 9, related joint as 3 and no related

TABLE 3. Reconstruction error (in unit of cm) for each joint by the method of Liu *et al.* [1].

Name of exercise with [6]		Left shoulder	Right shoulder	Left elbow	Right elbow	Left wrist	Right wrist
muscle-tightening deep breathing (Exercise 1)	All testing data	1.98	2.4	5.42	6.1	5.94	7.63
	Unreliable testing data	2.6	2.53	6.54	6.48	6.15	7.79
over the head pumping (Exercise 2)	All testing data	4.62	6.75	7.64	9.13	9.77	11.15
	Unreliable testing data	6.5	8.08	8.59	9.82	13.65	13.41
push down pumping (Exercise 3)	All testing data	3.22	3.47	8.56	8.78	9.86	10.26
	Unreliable testing data	5.89	6.46	10.84	10.65	11.93	12.68
horizontal pumping (Exercise 4)	All testing data	1.92	1.84	6.32	8.38	11.38	8.74
	Unreliable testing data	2.49	3.01	5.74	8.49	12.76	9.78

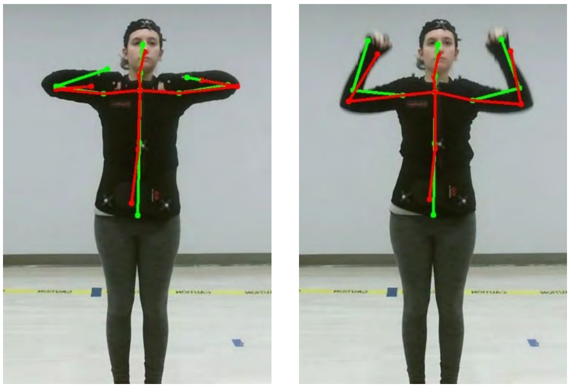

(a) (b)

FIGURE 8. The raw captured skeleton by Kinect (green) and the denoised skeleton by the proposed method (red) overlaid on the RGB image [1]. (a) horizontal pumping exercise; (b) push down pumping exercise.

joint as 0, then normalize the sum of the weighted value equal to 1.

5) DETECTION OF REPETITIONS AND ROBUSTNESS TO REPETITION VARIABILITY

Usually, in an exercise, the user is told to do several repetitions. For example, do ‘‘T-pose’’ to ‘‘ hand close to chest’’ and ‘‘ hand close to chest’’ to ‘‘T-pose’’ four times as shown in Fig. 10(b). In reality, users sometimes may forget how many repetitions they have already done, so they may do more or less repetitions than the reference sequence, which also can cause alignment error. We add a feature in the current system, which let the system choose what are the possible candidate subsequence after the current subsequence according to the user's motion input. For example, in Fig. 10(b), when the user is in key pose 2 (end of subsequence 3), the user can either do subsequence 2 again to go to key pose 3 or do subsequence 4 to go to key pose 4. When the user finishes subsequence 3, the system will compare the user's subsequent motion data with both reference subsequence 2 and reference subsequence 4, and find out the subsequence with the minimal matching error. Furthermore, the system detects the completion of one repetition upon the identification of subsequence 3 followed by subsequence 2. After the detection of each repetition, the system will display a message regarding how many more repetitions the user should do. At the end of this exercise, the system will give a friendly message to the user if the user did fewer or more repetitions.

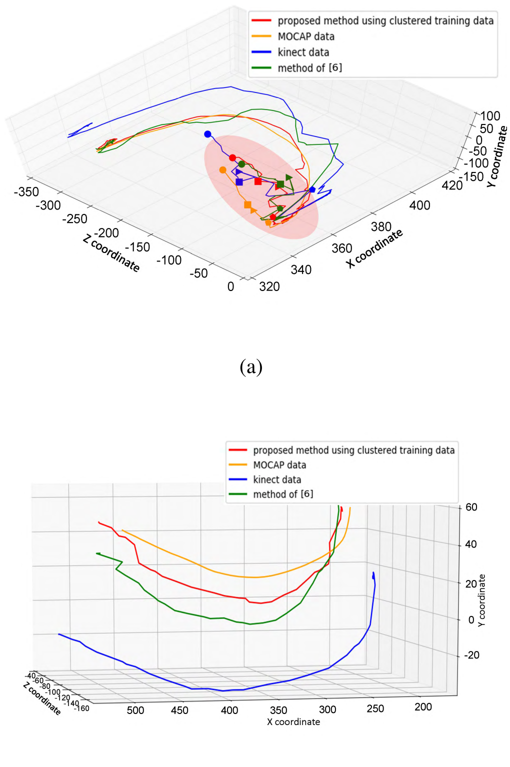
III. RESULTS AND DISCUSSION

In this section, we first present the motion sequence denoising results, and we then describe a prototype intervention system that we have developed that integrates all the components discussed in Sec. II.

A. EXPERIMENTAL RESULTS OF DENOISED MOTION SEQUENCE

Fig. 8 shows the skeleton captured by Kinect originally and the denoised skeleton using the proposed denoising method (consisting of four steps: standardization, clustered Gaussian Process regression, projection based on segment length constraint, inverse standardization). It is very clear to see that when the hands are close to the chest (Fig. 8(a)), the Kinect tracking result for the wrists can be very unreliable, even outside the human body. In Fig. 8(b), Kinect failed to accurately detect the positions of both elbows and wrists. The proposed method is able to correct these errors successfully. As a benchmark for evaluation, we have implemented the method described in [6] and applied it to our standardized data. We compare the reconstruction error of the denoising method of [6] and the proposed denoising method in Table 2. The reconstruction error is defined as the average Euclidean distance per joint between the Kinect-like converted MOCAP measured position (which we consider as the ground truth) and the denoised position from the Kinect data. We report the average error both over all test samples as well as over the unreliable test samples. We can see that the proposed method provides more accurate joint position estimation than [6]. Table 3 and Table 4 compares the reconstruction error in each joint. For the left and right shoulder, because they are never occluded and do not move much in the TOLF exercises, the reconstruction error is extremely small. On the other hand, the wrist joints have the largest error because when they are very often close to the chest, Kinect has trouble separating the wrists from the chest when observed from the front.

Fig. 9(a) shows the trajectory of the left wrist when a volunteer performs the horizontal pumping exercise. The volunteer closes her hands in front of her chest as shown in the pink ellipse region in the middle of the exercise. In this pose, the wrists are very close to the chest and the Kinect sensor cannot distinguish the difference in the depth of the wrists and the chest. This self-occlusion causes the Kinect sensor making the error in the estimated position of the wrists, which makes the joints trace unstable as shown on the blue line. The MOCAP position is shown on the orange line, which is



(a)

(b)

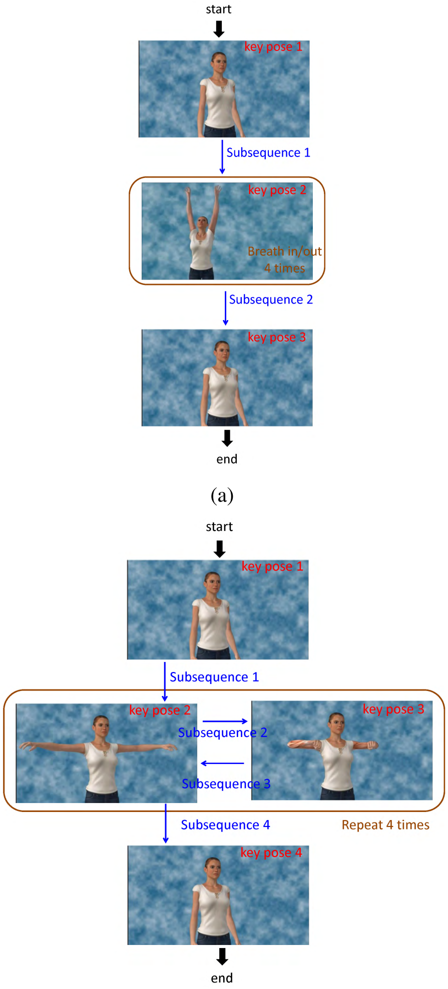
FIGURE 9. The raw captured joint trace by Kinect (green) and MOCAP (yellow), and the denoised trace by the proposed method (red) and the method of [6] (green) [1]. (a) The left wrist during the horizontal pumping exercise. Same markers (in shape) on different joint traces correspond to the same time, (b) The left elbow during the push down pumping exercise.

very smooth and can be treated as ground truth. Our denoised result is shown on the red line, which is very close to the ground truth and is relatively smooth compared with other results. Fig. 9(b) shows the trajectory of the left elbow when a volunteer is doing the push down pumping exercise. In this exercise the elbow is not blocked by other body part, so the trajectory is relatively smooth, but the Kinect estimated position has a consistent shift from the "true" position. Compared with the raw Kinect data and denoising result of method [6], our proposed method can eliminate most of the errors in the Kinect measurement and make the trajectory smooth and close to the ground truth (the MOCAP data).

B. PROTOTYPE INTERVENTION SYSTEM

1) OVERALL SYSTEM DESIGN AND PRELIMINARY RESULTS

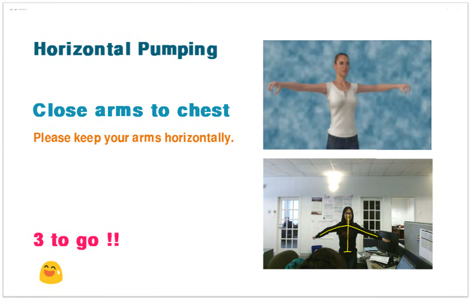
The various methods described in Sec. II, including motion sequence processing (standardization followed by denoising), temporal alignment, and for evaluating a denoised sequence against a reference sequence, have been integrated into a prototype system for guiding patients with risks for breast-cancer related lymphedema to perform a set of lymphatic exercises. All the processing steps can be completed in real time at 30 frames per second, with a



(a)

(b)

FIGURE 10. The subsequences and repetitions in different exercises. (a) Exercises 2: '"over the head pumping".(b)Exercise 4: "horizontal pumping".



Horizontal Pumping

Close arms to chest

Please keep your arms horizontally.

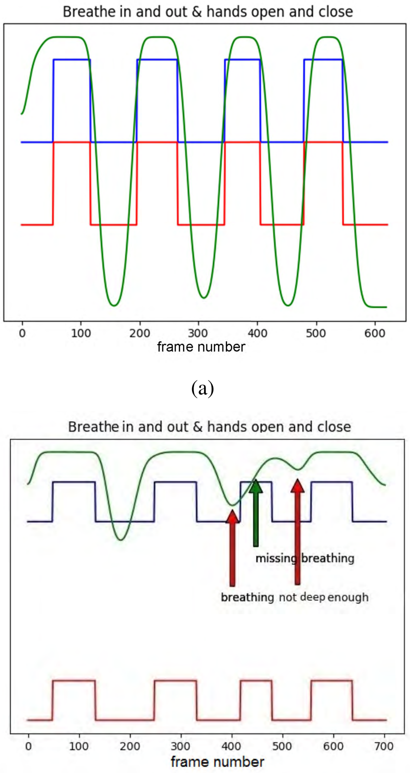
3 to go !!

FIGURE 11. Graphical Interface of the proposed system.

Windows 8 computer equipped with Intel i7-4720HQ CPU and 12 GB RAM. The interface of the prototype system is shown in Fig. 11. In this system interface, we provide several important information, which can guide users to do the exercises accurately. The user stands in front of a display and a Kinect sensor. The exercise avatar will be displayed at the upper right of the screen with both video and audio. The audio instruction in the avatar can help users to understand

TABLE 4. **Reconstruction error (in unit of cm) for each joint by the proposed method [1].**

Name of exercise with proposed method		Left shoulder	Right shoulder	Left elbow	Right elbow	Left wrist	Right wrist
muscle-tightening deep breathing (Exercise 1)	All testing data	0.14	0.14	1.57	2.01	2.14	2.78
	Unreliable testing data	0.21	0.21	3.77	3.19	2.41	3.4
over the head pumping (Exercise 2)	All testing data	0.13	0.13	1.95	1.94	3.08	3.33
	Unreliable testing data	0.17	0.17	4.38	4.14	7.4	7.75
push down pumping (Exercise 3)	All testing data	0.21	0.21	4.19	4.48	5.25	5.91
	Unreliable testing data	0.23	0.25	7.18	7.01	7.59	8.37
horizontal pumping (Exercise 4)	All testing data	0.18	0.18	2.68	2.76	5.08	5.33
	Unreliable testing data	0.25	0.44	3.81	4.21	7.77	7.59

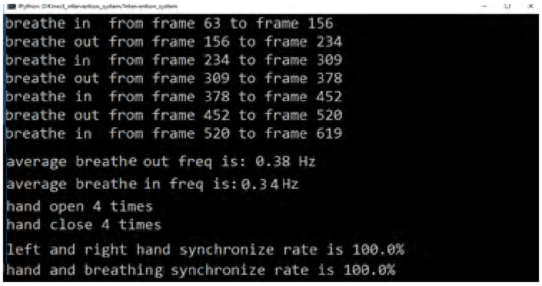


(a)

(b)

FIGURE 12. **Analysis results of breathing and hand status. In this figure, green curve describes the change of depth value in user's chest region, where the largest depth difference is 3.5 cm in this case. Blue and red curve represent the states of the left and right hand, respectively, where the higher value corresponds to "close" and small value corresponds to "open". (a) A case when a user breathes deeply and has perfect synchronization between the breathing state and the hand state. (b) A case when a user does not perform perfectly.**

what should they do. User's own video is displayed below the avatar video, with the denoised skeleton overlaid without perceivable delay. The left part of the screen shows the text-based feedback and instruction. For example, in the beginning of the exercise "horizontal pumping", it shows "Open arms to T-pose or close arms to chest" and then it displays "well done", or some suggestion for improvement after a user successfully finishes each subsequence. After the detection of each completed repetition, it will show how many more repetitions that the user has to do. At the end of the entire exercise, based on the internal analysis results, a message of compliment or suggestion for improvement will be displayed. We will discuss more detail for each exercise in next section.



```
breathe in  from frame 63 to frame 156
breathe out from frame 156 to frame 234
breathe in  from frame 234 to frame 309
breathe out from frame 309 to frame 378
breathe in  from frame 378 to frame 452
breathe out from frame 452 to frame 520
breathe in  from frame 520 to frame 619

average breathe out freq is: 0.38 Hz
average breathe in freq is:0.34Hz
hand open 4 times
hand close 4 times
left and right hand synchronize rate is 100.0%
hand and breathing synchronize rate is 100.0%
```

FIGURE 13. **Internal Analysis for exercise "over the head pumping".**

2) EXERCISE DETAILS AND DISCUSSION

So far, we have completed the algorithm designs for four exercises in the The-Optimal-Lymph-Flow^{TM} (TOLF) intervention. Here we provide details about Exercise 2, called "over the head pumping" as shown in Fig. 10(a). In this exercise, the the user is told to raise their hands up and hold in that position (subsequence 1). Then user needs to make four deep breathings. The user should close her hands while breathing in, and open hands while breathing out. The breathing detection is similar to the algorithm we use for the first exercise and the hands open/close state is detected by using the SDK of the Kinect sensor. After finishing 4 repetitions, the user should put down her hands on the sides to finish the exercise (subsequence 2). By analyzing the captured depth data and the hands' state data, the system will derive pertinent information, including breathing frequency as shown in Fig. 12, and whether the breathing is synchronized with the corresponding hand state as shown in Fig. 13. If a user does not breathe deep enough or forget to breathe during the exercise period, or if the hand state is not synchronized properly with the breathing state, the system will send out a friendly message to remind the user to breathe in and out more deeply, or to close/open the hands when breathing in/out. In this exercise, wrist joints play an important role, so we assign high weights for the wrist joints than for the other joints. Appendix B provides details for the other three exercises.

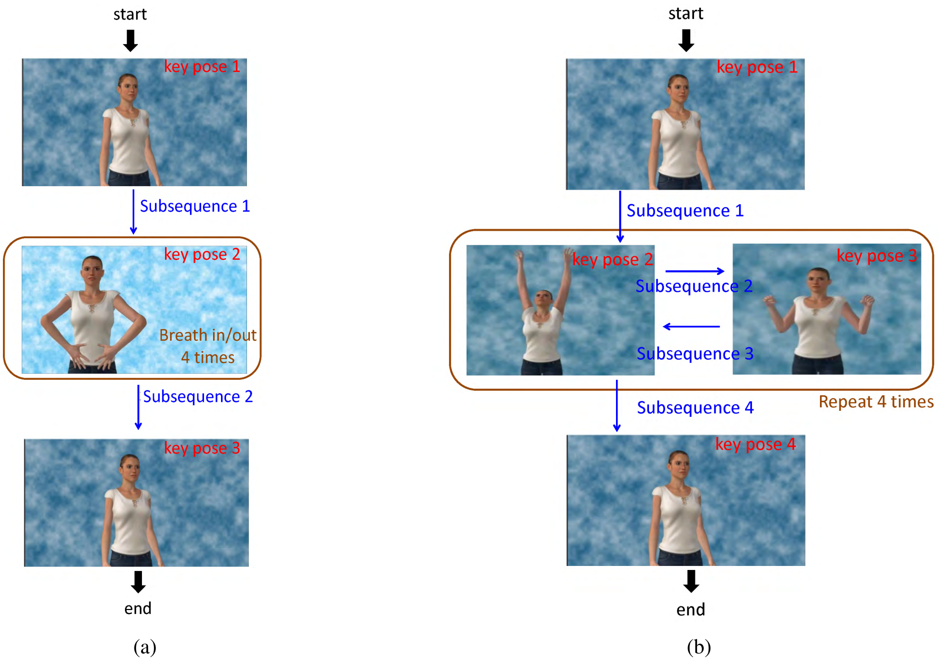
IV. CONCLUSION

We have developed an exercise guidance system, which can automatically detect whether a user is performing a set of exercises properly, based on the depth and joint positions captured by a Kinect sensor. The first main technical innovation is a clustered Gaussian Process regression model for denoising the Kinect joint measurements. We map joint traces

TABLE 5. The-Optimal-Lymph-Flow (TOLF) exercise to promote lymph flow [16].

TOLF Exercise	Rationales	Frequency & Situations
Muscle-Tightening Deep Breathing	• The whole body lymph fluid has to be drained through the lymphatic ducts above the heart. Muscle-tightening-deep-breathing stimulates lymphatic ducts and help lymph fluid drain. • Lymph fluid drains when muscles move. Muscle-tightening-deep-breathing creates the whole body muscle movements that create muscle milking and pumping action and help to drain lymph fluid.	• At least twice a day in the morning & at night before brushing teeth or as much as the patient wants throughout the day. • Air-Travel: before take-off and after landing. • Sedentary life-style: At least every 4 hours.
Muscle-Tightening Pumping	• Muscle-tightening pumping exercises create arm muscle pumping. This helps lymph fluid flow and decreases the fluid build-up in the arms. • Muscle-tightening pumping exercises build the arm muscle that helps lymph fluid flow and drain.	• At least twice a day in the morning & at night before brushing teeth or as much as the patient wants throughout the day. • Air-Travel: before take-off and after landing. • Sedentary life-style: At least every 4 hours.
Shoulder Exercises	• Improved limb mobility after surgery facilitates local muscle movements that create muscle milking and pumping that help local limb lymph fluid flow and drain.	• One week after surgery if there are no surgical drains or after the surgical drains are removed. • At least twice a day until limb functions are returned to normal. • Whenever limb mobility is limited throughout the recovery.
Large Muscle Exercise: Walking, Marching at home, Dancing, Swimming, Yoga, Tai Chi.	• Large muscle exercises create muscle milking and pumping to promote overall body lymph fluid flow and drain.	• At least 10-minutes daily. • Air-Travel: get up and walk around for over 4 hours flight. • Sedentary life-style: Get up and walk at least every 4 hours.

captured by both a Kinect sensor and a MOCAP system to a standardized domain in order to circumvent the difficulty caused by the high variance of the joint positions due to the variety of human body sizes and learn the regression model in this domain. To reduce the computation with the regression model, we cluster all the training samples into a small number of groups and use the cluster centroids as the reduced training set. The second main contribution is a modified dynamic time warping algorithm that can automatically detect the end of each exercise subsequence while simultaneously aligning the detected subsequence with a reference subsequence. This enables accurate evaluation of the user's movement against the reference sequence for each exercise subsequence and instantaneous feedback while a user is performing the exercise. Our experiments show that the proposed denoising method can effectively correct the errors in Kinect measurements due to self-occlusion, and performs better than a benchmark system. The overall system can accurately align



(a) (b)

FIGURE 14. The subsequences and repetitions in different exercises. (a) Exercise 1: "muscle-tightening deep breathing". (b)Exercise 3: "push down pumping".

a user's motion sequence with the reference sequence, and evaluate the "correctness" of user's movements in real time, enabling instantaneous feedbacks to the user while the user is performing an exercise. The Kinect-based system has potential to be used for similar health problems related to lymphatic health as well as science, art and entertainment involving body movements.

ACKNOWLEDGMENT

The authors would like to thank Dr. Winslow Burleson and Dr. Stephen Jeremy Rowe from NYU-X lab for assistance with recording the motion sequences, and to all the volunteers who participated in the recording of training data. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

APPENDIX

A. THE RATIONALE OF TOLF EXERCISE

TOLF exercise program was designed for breast cancer patients to promote lymph flow for upper limbs and body. The rationale for each TOLF exercise can be found in Table 5. TOLF exercise program is consistent with the recommendations of the American Cancer Society [28] for breast cancer patients regarding seeking help from health professionals for exercise after breast cancer treatment. TOLF exercise has been delivered to patients in-person with trained nurses. The Kinect system will help nurses to deliver the TOLF exercise accurately and help patients to perform TOLF exercise at home with accuracy and motivation. The duration and repetitions for each TOLF exercise can be personalized based on patients' ability and preference.

B. EXERCISE DETAILS AND DISCUSSION

The-Optimal-Lymph-Flow^{TM} intervention consists of seven exercises. So far, we have completed algorithm design for four exercises. We have described Exercise 2 in Sec.III-B.2. Here we provide details for the other three exercises. Exercise 1 is called "muscle tightening deep breathing" as shown in Fig. 14(a), where the user needs to tighten her body muscle, raise her hands to the belly position (subsequence 1), stay in this pose and make four deep breathings, and finally put down her hands to finish the exercise (subsequence 2). In this exercise, we mainly focus on whether the user performs deep breathing appropriately in pose 2 (after detecting the end of subsequence 1). This is accomplished by evaluating the depth variation in the chest region. In the evaluation state, the system automatically identifies a small region in the center of the chest in the first frame according to the relative position of two shoulders and spine-middle and finds the average depth value in that region using the depth map generated by the Kinect sensor. Then, for the following frames, the system finds the difference between the mean depth value in the chest region in the current frame and that in the first frame. This difference is defined as the depth value for the current frame. We draw the depth variation as a function of time (see Fig. 12(a)), and find the local minimum (corresponding to breath in) and maximum (corresponding to breath out). If the depth difference between the local maximum and minimum is less than a certain threshold, the system will send a reminder to the patient to breathe deeper. In this exercise, all the joints share the same importance, so the weights for determining the average gradient magnitude in Eq. 3 are the same.

Exercise 3 is called "push down pumping" as shown in Fig. 14(b). First, the user should raise her hands as high as possible with hands open (subsequence 1). Then, the user should start breathing in with hand closed and at the same time push down two elbows (subsequence 2), with each arm forming a "V-shape" at the end of subsequence 2. After staying for a while in this pose, the user should raise her hands again to as high as possible while breathing out and open hands (subsequence 3). The user repeats this four times and finally puts down the hands (subsequence 4). In this exercise, we mainly check whether the positions of wrists, elbows and shoulders are in a straight line in pose 2, and whether the positions of wrists, elbows and shoulders form a V-shape rather than L-shape. The system analyzes the user's performance based on the detected positions of the wrists, elbows and shoulders at the end of each subsequence and gives the user appropriate feedback. In this exercise, wrists and elbows are more important than other joints, so the weights for wrists and elbows are higher than for other joints.

Exercise 4 is called "horizontal pumping", as shown in Fig. 14(c). A user first raises her hands to the horizontal position to form a "T-pose" with hands open (subsequence 1). Then the user slightly bends her elbows to move her hands to the chest region while closing her hands and breathing in slowly (subsequence 2). Next, the user moves back to the "T-pose" while opening the band and breath out (subsequence 3). After repeating subsequences 2 and 3 four times, the user puts down her hands on the sides (subsequence 4) and finishes this exercise. In this exercise, the system checks the positions of wrists, elbows and shoulders form a horizontal line during the "T-pose", and whether the hand close/open state is correct. In this exercise, wrist positions are more important than the elbow positions, which are more important than other joints. Therefore, we assign the highest weights for the wrists, and second highest weights for the elbows.

REFERENCES

[1] A.-T. Chiang, Q. Chen, S. Li, Y. Wang, and M. Fu, "Denoising of joint tracking data by Kinect sensors using clustered Gaussian process regression," in *Proc. 2nd Int. Workshop Multimedia Pers. Health Health Care*, 2017, pp. 19–25.

[2] *Microsoft Kinect v2*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.xbox.com/en-US/xbox-one/accessories/kinect

[3] *Intel realsense*. Accessed: Jul. 21, 2017.[Online]. Available: https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

[4] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for Microsoft Kinect," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1357–1369, Oct. 2013.

[5] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 188:1–188:12, Nov. 2012.

[6] Z. Liu, L. Zhou, H. Leung, and H. P. Shum, "Kinect posture reconstruction based on a local mixture of Gaussian process models," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 11, pp. 2437–2450, Nov. 2016.

[7] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 449–456.

[8] S. R. Tripathy, K. Chakravarty, A. Sinha, D. Chatterjee, and S. K. Saha, "Constrained Kalman filter for improving Kinect based measurements," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[9] D. S. Alexiadis, P. Kelly, P. Daras, N. E. O'Connor, T. Boubekeur, and M. B. Moussa, "Evaluating a dancer's performance using Kinect-based skeleton tracking," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 659–662.

[10] A. Yurtman and B. Barshan, "Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units," in *Information Sciences and Systems*. Cham, Switzerland: Springer, 2013, pp. 305–314.

[11] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[12] C.-J. Su, C.-Y. Chiang, and J.-Y. Huang, "Kinect-enabled home-based rehabilitation system using dynamic time warping and fuzzy logic," *Appl. Soft Comput.*, vol. 22, pp. 652–666, Sep. 2014.

[13] W. Wei, Y. Lu, C. D. Printz, and S. Dey, "Motion data alignment and real-time guidance in cloud-based virtual training system," in *Proc. Conf. Wireless Health*, 2015, Art. no. 13.

[14] *Optitrack*. Accessed: Sep. 21, 2017. [Online]. Available: http://optitrack.com/

[15] H. G. Armstrong, "Anthropometry and mass distribution for human analogues, volume I: Military male aviators," USAARL Rep. 88-5, 1988.

[16] M. R. Fu *et al.*, "Proactive approach to Lymphedema risk reduction: A prospective study," *Ann. Surgical Oncol.*, vol. 21, no. 11, pp. 3481–3489, 2014.

[17] M. R. Fu *et al.*, "Usability and feasibility of health IT interventions to enhance self-care for Lymphedema symptom management in breast cancer survivors," *Internet Interventions*, vol. 5, pp. 56–64, Sep. 2016.

[18] M. R. Fu *et al.*, "mhealth self-care interventions: Managing symptoms following breast cancer treatment," *Mhealth*, vol. 2, p. 28, Jul. 2016.

[19] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*. Berlin, Germany: Springer, 2004, pp. 63–71.

[20] Y. Cao, M. A. Brubaker, D. J. Fleet, and A. Hertzmann, "Efficient optimization for sparse Gaussian process regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1097–1105.

[21] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1257–1264.

[22] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[23] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proc. SIAM Int. Conf. Data Mining*, 2004, pp. 11–22.

[24] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.

[25] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 491–502.

[26] S. Soheily-Khah and P.-F. Marteau. (2017). "Sparsification of the alignment path search space in dynamic time warping." [Online]. Available: https://arxiv.org/abs/1711.04453

[27] V. M. Kumar and D. S. H. Thipesh, "Robot ARM performing writing through speech recognition using dynamic time warping algorithm," *Int. J. Eng. Trans. B, Appl.*, vol. 30, no. 8, pp. 1238–1245, 2017.

[28] *American Cancer Society*. Accessed: Apr. 10, 2018. [Online]. Available: https://www.cancer.org/