Human Action Understanding and Movement Error Identification for the Treatment of Patients with Parkinson's Disease

Wenchuan Wei
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA, USA
w8wei@eng.ucsd.edu

Carter McElroy
Department of Rehabilitation Services
University of California, San Diego
La Jolla, CA, USA
cjmcelroy@ucsd.edu

Sujit Dey
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA, USA
dey@ece.ucsd.edu

***Abstract*—Traditional physical therapy treatment for patients with Parkinson's disease (PD) requires regular visits with the physical therapist (PT), which may be expensive and inconvenient. In this paper, we propose a learning-based personalized treatment system to enable home-based training for PD patients. It uses the Kinect sensor to monitor the patient's movements at home. Three physical therapy tasks with multiple difficulty levels are selected by our PT co-author to help PD patients improve balance and mobility. Criteria for each task are carefully designed such that patient's performance can be automatically evaluated by the proposed system. Given the patient's motion data, we propose a two-phase human action understanding algorithm TPHAU to understand the patient's movements. To evaluate patient performance, we use Support Vector Machine to identify the patient's error in performing the task. Therefore, the patient's error can be reported to the PT, who can remotely supervise the patient's performance and conformance on the training tasks. Moreover, the PT can update the tasks that the patient should perform through the cloud-based platform in a timely manner, which enables personalized treatment for the patient. To validate the proposed approach, we have collected data from PD patients in the clinic. Experiments on real patient data show that the proposed methods can accurately understand patient's actions and identify patient's movement error in performing the task.**

Keywords—Parkinson's Disease, Physical Therapy, Action Segmentation, Hidden Markov Model, Support Vector Machine

I. Introduction

Parkinson's disease (PD) is one of the most common neurodegenerative movement disorders, especially in the elderly. More than 10 million people worldwide are living with PD. In the US, about 60,000 people are diagnosed with PD each year [1]. The major motor symptoms of PD include tremor, rigidity, and postural instability. Treatment for PD patients includes medication and physical therapy. In the traditional physical therapy treatment procedure (see Fig. 1(top)), a physical therapist (PT) decides the initial training tasks that the patient should practice in an initial mobility and balance evaluation session and instructs the patient on how to perform the tasks correctly. After the initial evaluation session, the patient is expected to practice the training tasks at home. In the next visit, the PT inspects the patient's performance on the training tasks and progresses the tasks based on the patient's performance.

However, there are some problems with the traditional treatment procedure. First, patient's performance and conformance on the training tasks cannot be tracked at home. In the traditional physical therapy session, the PT needs to carefully inspect the patient's movements and identify his/her errors in performing the task. Practicing the task with incorrect technique is not only ineffective for motor learning, it may also cause injury due to the impaired mobility of PD patients. However, patient's performance cannot be tracked at home without the supervision of the PT. King et al. has shown poor outcomes with home-based exercise programs for PD patients [2]. To address this problem, some home-based automatic training systems have been developed, where patient's movements are captured by motion capture sensors, including wearable sensors [3, 4] and camera-based sensors [5, 6]. These training systems can motivate the patients and monitor their movements at home. However, patient's performance cannot be accurately evaluated and no feedback is provided for the patients and the PT. Second, the training tasks cannot be updated in a timely manner before the patient's next visit with the PT, even with patient's significant progress or regress. Continuing to practice the same training task, which may not be suitable any more for the current state of the patient, could reinforce motor learning in a negative way. Third, PT's instructions and assessments are based on their experience, sometimes with subjective bias.

In this paper, we propose a learning-based treatment system for PD patients in Fig. 1(bottom). Three physical therapy tasks with multiple difficulty levels (shown in Table I) are selected by our PT co-author to help PD patients improve balance and mobility. The patient trains himself/herself at home using the cloud-based system we proposed in [7], where the Microsoft Kinect sensor [8] is used to capture the patient's movements and avatars are created to provide visual feedback instruction. Given the patient's motion data, we propose a two-phase human action understanding (TPHAU) algorithm to understand the patient's movements, including how many repetitions the patient has done, and which sub-action the patient is doing at any given frame. Furthermore, we propose a Support Vector Model (SVM) based method to identify the patient's error in performing the task. Therefore, the patient's error can be reported to the PT, who can be potentially offline or remote, to supervise the patient's performance and conformance on the training tasks. Moreover, the PT can decide change in the training tasks according to the

2575-2634/18/$31.00 ©2018 IEEE
DOI 10.1109/ICHI.2018.00028

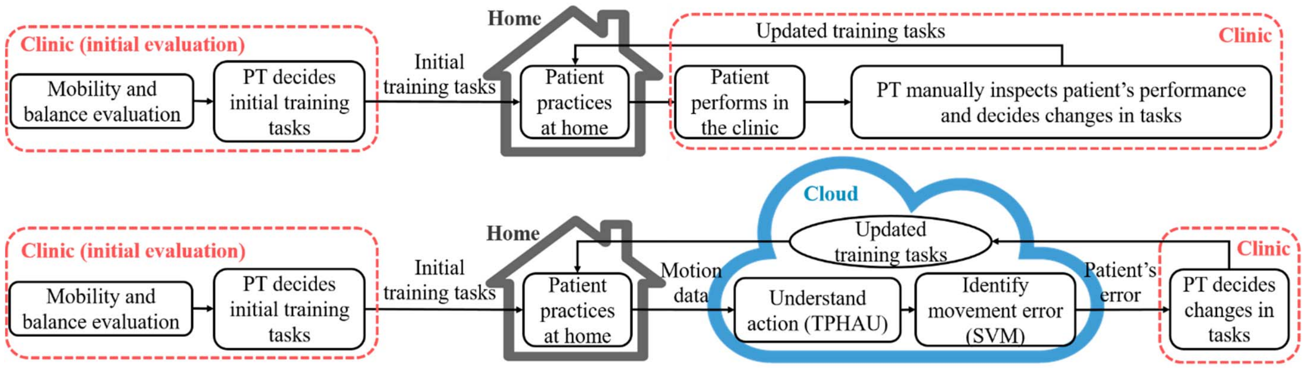

Fig. 1. Procedure for the treatment of PD patients. Top: traditional procedure. Bottom: the proposed treament system.

patient's progress and update the tasks remotely through the cloud-based platform in a timely manner, which enables personalized treatment for the patient. Besides offering timely and personalized care, the proposed treatment system has the potential of significantly reducing cost, and can be particularly useful for remote care. It reduces PT's subjective bias in evaluating patient's performance by using machine intelligence. To validate the effectiveness of the proposed methods, we have collected real patient data in the Neurological Rehabilitation Clinic, UC San Diego Health. Experiments on the collected patient data show the accuracy of the proposed human action understanding and error identification methods.

The remainder of this paper is organized as follows: Section II reviews related work about home-based training systems for PD patients and human action understanding techniques. In Section III, we introduce the training tasks selected by the PT for PD patients and the criteria used to identify errors. Section IV and Section V propose the TPHAU algorithm for human action understanding and the SVM-based patient error identification method. Section VI presents the experimental results. Section VII concludes the paper and discusses future work.

II. Related Work

In recent years, more and more home-based automatic training systems are being developed for the treatment of patients with balance and/or mobility problems. Esculier et al. [9] design a home-based balance training system using Wii Fit with Balance Board and show its ability in improving static and dynamic balance, mobility and functional abilities of PD patients. Finkelstein et al. [10] develop a home-based telerehabilitation system for patients with mobility limitations. Jeong et al. [11] introduce a physical telerehabilitation system and experiments prove high level of acceptance of this system by patients with significant mobility disability. However, patient's movements cannot be accurately monitored in these systems. To solve this problem, some training systems use motion capture sensors, including wearable sensors and camera-based sensors, to capture patient's movements during the home sessions. Chen et al. [3] propose a platform to enable home monitoring of PD patients using wearable sensors. Pan et al. [4] develop a cloud-based mobile application that enables home-based assessment and monitoring of major PD symptoms. In [4], the smartphone 3D accelerometer is used to collect patient's motion data, and the smartphone is mounted on the patient's hand or ankle with a strap. However, wearable sensors attached on the body may cause extra burden to PD patients due to their impaired mobility. Therefore, camera-based sensors were considered more convenient in monitoring the movements of PD patients. Galna et al. [12] prove the high accuracy of the Kinect sensor in measuring clinically relevant movements in PD patients. Pompeu et al. [6] and Galna et al. [5] design two game-based training systems using Kinect and prove their feasibility and safety for PD patients. However, the game-based training systems are not able to carefully control the patient's movements and cannot provide accurate feedback on the patient's performance. Lin et al. [13] design a Kinect-based rehabilitation system to assist patients with movement disorders and balance problems. However, the performance evaluation method proposed in [13] fails to consider the patient's reaction delay (i.e., delay by the patient to follow the standard movements) as the method simply compares the patient's movements with the standard movements frame by frame.

In our proposed treatment system for PD patients, we use the Kinect-based training system reported in [7] but redesign the user performance evaluation model. Although the Gesture-based Dynamic Time Warping algorithm proposed in [7] addresses the motion data misalignment problem caused by human reaction delay and network delay, it evaluates the patient's performance by measuring the similarity between his/her motion data with a PT template, which may be subject to low accuracy when the patient's performance is poor. The template provided by the PT may also contain PT's subjective bias. In comparison, the TPHAU algorithm proposed in this paper does not need any template. It addresses the delay problem and evaluates patient's performance accurately since the Hidden Markov Model (HMM) used in TPHAU can compensate for the temporal variation of patient's motion data. Besides, the training tasks discussed in [7] (e.g., leg lift) are relatively simple and the criteria apply to the entire task, and hence cannot be applied to real tasks for PD patients considered in this paper. For example, the criterion "keep the right knee straight" used in [7] for task "leg lift" means that the patient should keep the right knee straight for the entire time when he/she is doing this exercise. In comparison, the training tasks selected for PD patients, as described in this paper, are more complex and are based on actual balance and agility programs that have been developed for this patient population. The tasks

TABLE I. TASKS AND VARIATIONS. FROM LEFT TO RIGHT: SQ, FL, BL.

Variations of SQ	Hand support	Squatting angle
SQ1	Yes	Small
SQ2		Large
SQ3	No	Small
SQ4		Large

Variations of FL	Hand support	Length of step
FL1	Yes	Small
FL2		Large
FL3	No	Small
FL4	Arms up	Large

Variations of BL	Hand support	Length of step
BL1	Yes	Small
BL2	Step back with hand support, then take hands off	Large
BL3	No	Small
BL4		Large

for PD patients require more fine-grained application of criteria: different criteria applied to different subsets of sub-actions of a task (discussed in Section III-B), as opposed to all the criteria applied to the entire task (i.e., including all the sub-actions) assumed in [7]. Therefore, action understanding is needed to detect patient's sub-actions in performing the task.

There has been numerous research in the field of human action understanding. Most of them are focused on recognizing human actions from videos, including common RGB videos and RGB-D videos recorded by Kinect. Generally, these studies can be divided into two categories: action recognition and action detection/segmentation. Action recognition refers to the classification of an action from a given video into some templates. In action recognition, the time and space range of the action are known. Xia et al. [14] develop an approach to recognize human actions using the histogram of joint locations from Kinect depth maps. Sung et al. [15] propose a two-layered maximum entropy Markov model to recognize human actions from RGB-D videos. In our proposed treatment system for PD patients, action recognition is not needed since the task that the patient currently performs is known to us. What we need is to detect/segment the sub-actions that the patient performs. Action detection/segmentation refers to locating actions of interest in space and/or in time. Pirsiavash et al. [16] propose an algorithm to detect action segments from long video streams. Wu et al. [17] present an unsupervised learning algorithm for video segmentation and recognition from RGB-D videos recorded by Kinect. Most of these action detection/segmentation techniques are focused on the detection of long action segments. In [16] and [17], a detected segment is considered correct if the overlap between it and the ground-truth action segment is more than 40%, as this threshold is consistent with visual inspection. However, the sub-actions discussed in this paper are much shorter in time length and closer to each other (the pause between adjacent sub-actions is negligible), which makes the segmentation between them much more challenging (i.e., frames around the boundary between two sub-actions may be incorrectly segmented). In this paper, we propose the TPHAU algorithm to accurately detect patient's sub-actions in performing the task, which will be discussed in Section IV.

III. KINECT-BASED AUTOMATIC TRAINING SYSTEM FOR PATIENTS WITH PARKINSON'S DISEASE

In this section, we will first introduce the physical therapy tasks selected by our PT co-author for PD patients. Then we will discuss how the proposed Kinect-based training system can be used for self-training on these tasks at home and how it can identify patient's error in performing the task automatically. To avoid confusion, we would like to clarify the definition of four terms: task, movement/action, repetition, and sub-action. Task is an exercise designed by the PT to train patients. Movement/action is the execution/performance of the task by a patient, which may contain one or multiple repetitions. Each repetition can be further divided into several sub-actions, which will be introduced in Section III-B.

A. Tasks and Variations

Based on the work of King and Horak that describes sensorimotor agility training and shows improvements in balance and mobility in PD patients [18], our PT co-author has selected three balance/agility based tasks: squat (SQ), forward lunge (FL) and backward lunge (BL). For each task, four variations with different difficulty levels are designed (see Table I). The difficulty level increases from level 1 to level 4. During a traditional physical therapy session, a patient performs one variation of each task, and the PT identifies the patient's error and decides how to update the tasks for the patient (see Fig. 1). For example, if the patient performs excellently on the current SQ2, the PT may instruct him/her to progress to SQ3. On the contrary, a patient who makes serious errors on the current SQ2 may need to go back to SQ1. PT's evaluation is based on self-designed criteria for each task. Criteria are based on different sub-actions of the movements, which will be introduced in Section III-B.

B. Sub-actions and Criteria

For each physical therapy task, patient's movements can be divided into several sub-actions. For example, movements in FL include: 1) stand, 2) step forward, 3) maintain balance control, 4) return to the original position, 5) stand. Therefore, we define five sub-actions $S_1 \sim S_5$ in Table II, which apply to all the tasks considered for PD patients: SQ, FL and BL.

TABLE II. SUB-ACTIONS OF PATIENT'S MOVEMENTS

Sub-action	Patient's movements
S_1	Standing
S_2	Movement initiation: try to reach the target position
S_3	Balance hold: maintain balance control
S_4	Return to the original position
S_5	Standing

Criteria of a task (i.e., the rules for evaluating the patient's performance) are carefully designed by our PT co-author such that the patient's performance can be evaluated by the proposed system in an automated and quantified way. A task criterion is applicable to one or more sub-actions of the task. Table III shows the criteria defined by our PT co-author and the applied sub-actions for SQ, FL, and BL. For example, in FL, the patient should keep the back knee straight only in S_2 and S_3.

In the Kinect-based training system, the Kinect sensor captures 25 joints of the human skeleton with 3-D coordinates for each joint [8]. To enable automatic error identification, we first need to translate PT's criteria into some Kinect-captured quantities (KCQs). KCQs are quantities that can be derived from the joint coordinates captured by Kinect. In this paper, we define the following six KCQs for the three tasks. (Considering the difference in body size, we use normalized quantities, e.g., angles and normalized length of step.)

Thigh Angle (ThA): the angle between the thigh and the vertical direction. In SQ, we use the average of left and right thigh angles to represent the squatting angle.

Trunk Angle (TrA): the angle between the trunk and the vertical direction. It represents the forward-leaning angle in SQ and can be used to check whether posture is tall in FL.

Trunk-Leg Angle (TrLA): the angle between the trunk and the back leg. In BL the patient should slightly lean forward thus keep the trunk parallel with the back leg.

Knee Angle (KA): the angle between the thigh and the shank, representing whether the knee is straight.

Normalized Length of Step (NLoS): the distance between the two feet, normalized by the length of leg.

Shank Angle (SA): the angle between the shank and the vertical direction, representing whether the shank is vertical.

Fig. 2 illustrates these KCQs. KCQs that are used in more than one task (e.g., NLoS used in FL and BL) are shown in only one task for simplicity. The target value of each KCQ shown in Table III is either defined by the PT (e.g., KA: 180°) or derived from PT's demonstration (e.g., ThA: 49° for small angle and 67° for large angle).

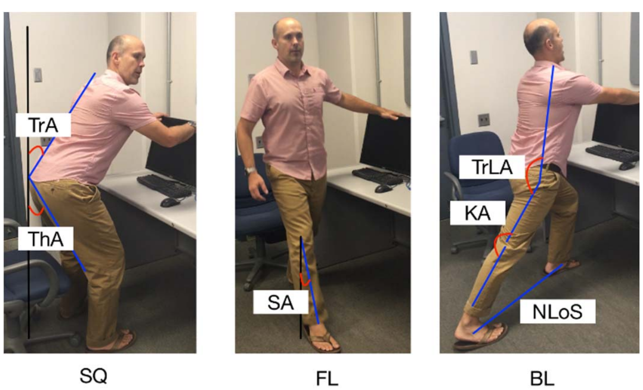

Fig. 2. Tasks and KCQs. From left to right: SQ, FL, BL.

Given the KCQs, patient's performance can be evaluated automatically by checking the KCQs in the applied sub-actions. In Section IV, we will introduce how to segment sub-actions in patient's movements, i.e., understand which sub-action the patient is in for a given time/frame, from his/her motion data.

IV. Human Action Understanding

In this section, we propose an algorithm based on Hidden Markov Model (HMM) to detect sub-actions in patient's movements. Section IV-A introduces the two HMM models we propose for single repetition and multiple repetitions of a task. Section IV-B discusses the selection of HMM features. In Section IV-C, we propose the TPHAU algorithm which uses two different HMM models in two phases to segment sub-actions in patient's movements.

A. Hidden Markov Model

HMM is a statistical Markov model that assumes the system to be a Markov process with hidden states. It is widely used in speech recognition [19]. A HMM model is characterized by the following elements.

- Hidden state set: $\{S_1, S_2, \cdots, S_N\}$ where N is the number of hidden states. Note that one state in the HMM model represents a sub-action in patient's movements, thus we use the same symbol S_i for both.
- HMM feature: it is the quantity we observe in the HMM model. The HMM feature can be one-dimensional or multi-dimensional. For example, we can choose NLoS as the HMM feature for task FL. Selection of the HMM features will be discussed in Section IV-B.
- Observation sequence $O = \{O_1, O_2, \cdots, O_T\}$, where O_t is the value of the HMM feature at frame *t* and *T* is the total number of frames. For example, if we choose NLoS as the HMM feature for task FL, O_t is the value of NLoS at frame *t*.
- Hidden state sequence $Q = \{q_1, q_2, \cdots, q_T\}$, where q_t is the hidden state at frame *t*. q_t can be any state from the hidden state set. For example, $q_t = S_i$ means that the patient is performing the sub-action S_i in frame *t*. The hidden state sequence can be used to segment patient's sub-actions.
- State transition probability matrix $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i],\ 1 \le i, j \le N. \quad (1)$$

  It represents the probability of transferring from state/sub-action S_i to state/sub-action S_j.
- Emission probability $b_j(X)$, which is the probability of obtaining the observation X under state S_j. $b_j(X)$ is defined as

$$b_j(X) = P[O_t = X \mid q_t = S_j],\ 1 \le j \le N. \quad (2)$$

  For example, if we choose NLoS as the HMM feature for the task FL, $b_2(0.8)$ represents the probability of obtaining $NLoS = 0.8$ at S_2. In discrete-HMM (i.e., observation is discrete), the emission probability is defined in a matrix. In continuous-HMM (i.e., observation is continuous), we can either quantize the continuous data via codebooks, or use a continuous probability density (e.g., Gaussian distribution). In the continuous case, $b_j(X)$ is not a probability as it can be larger than 1. However, it represents the relative likelihood of obtaining an observation under a certain state.

TABLE III. PT-DEFINED CRITERIA, KCQS AND APPLIED SUB-ACTIONS. FROM LEFT TO RIGHT: SQ, FL, BL.

SQ: PT's Criterion	KCQ	Applied sub-actions
Sit hips back towards a chair	ThA: 49° (small), 67° (large)	S_3
Lean forward	TrA: 22° (small), 27° (large)	S_3

FL: PT's Criterion	KCQ	Applied sub-actions
Keep the back knee straight	KA (back leg): 180°	S_2, S_3
Keep the posture tall	TrA: 0°	S_2, S_3, S_4
Length of step	NLoS: 0.47 (small), 0.79 (large)	S_3
Keep the front shank vertical	SA (front leg): 0°	S_3

BL: PT's Criterion	KCQ	Applied sub-actions
Keep the back knee straight	KA (back leg): 180°	S_3
Keep the trunk parallel with the back leg	TrLA: 0°	S_2, S_3, S_4
Length of step	NLoS: 0.48 (small), 0.78 (large)	S_3
Keep the front shank vertical	SA (front leg): 0°	S_2, S_3

- Initial state distribution π_i where

$$\pi_i = P[q_1 = S_i],\ 1 \le i \le N. \tag{3}$$

We define a 5-state HMM model HMM-S (where 'S' represents single) for the single repetition of a training task in Fig. 3. Each state represents one sub-action in the movements. In this left-to-right model, the initial state is fixed to be S_1. Therefore $\pi_1 = 1, \pi_i = 0\ (i > 1)$.

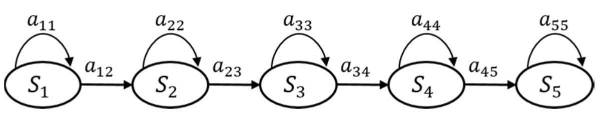

Fig. 3. HMM-S: the HMM model for single repetition.

In a typical physical therapy session, the PT may instruct the patient to perform each task for multiple repetitions. Therefore, after the fifth sub-action S_5 (i.e., standing, see Table II), the patient goes back to the second sub-action S_2 (i.e., movement initiation). In this case, S_1 and S_5 can be combined to define the following HMM model HMM-M (where 'M' represents multiple) for multiple repetitions on the task in Fig. 4. Although it is not a left-to-right model, we can still conclude that the patient starts from S_1, thus $\pi_1 = 1, \pi_i = 0\ (i > 1)$.

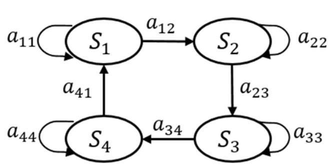

Fig. 4. HMM-M: the HMM model for multiple repetitions.

1) Estimation of Model Parameters

Parameters of a HMM model include the transition matrix A, the emission probability $b_j(X)$, and the initial state distribution π_i. Given any finite observation sequences $O_{training}$ as training data, there is no optimal way of estimating the model parameters $\lambda = \{A, b_j(X), \pi_i\}$. In speech recognition, the Baum-Welch method [20] and gradient techniques [21] are often used to choose model parameters such that $P(O_{training} \mid \lambda)$ is locally maximized (i.e., optimize the model parameters λ so as to best describe how the given observation sequences $O_{training}$ are produced by the model). Then based on all the HMM models $\lambda_1, \lambda_2, \cdots, \lambda_K$ for the speech vocabularies (with one HMM model for one speech vocabulary), speech recognition can be further achieved by finding the model λ_i that maximizes $P(O_{test} \mid \lambda_i)$ for any new observation O_{test}.

In the proposed HMM models for physical therapy tasks, there is no need to classify a new movement into an optimal task as the task that the patient is performing is known to us. Our goal is to infer the hidden state sequence *Q* from the observation sequence *O*, then segment the interval of each sub-action according to the hidden state sequence. Therefore, we choose supervised learning to train the model parameters. For all the training data, five sub-actions in the movements (see Table II) are manually segmented based on the patient's motion video and motion data. (Note that for HMM-M, S_1 includes the manually-labelled S_1 and S_5.) The transition probability matrix is calculated as

$$a_{ij} = \frac{\text{number of transitions from } S_i \text{ to } S_j}{\text{number of transitions from } S_i},\ 1 \le i, j \le N. \tag{4}$$

For the emission probability, we use the Gaussian Mixture Model (GMM) as

$$b_j(X) = \sum_{c=1}^{C} w_{jc} \mathcal{N}(\mu_{jc}, \Sigma_{jc}), \tag{5}$$

where *C* is the number of mixture components, $w_{jc}, \mu_{jc}, \Sigma_{jc}$ are the weight, mean, and covariance of the *c-th* Gaussian component. Parameters of GMM are estimated from the training data using the Expectation-Maximization (EM) algorithm [22]. The GMM model of each sub-action/state is trained separately using the motion data in that state.

2) Estimation of Hidden states

Given an observation sequence *O* and the model parameters λ, our goal is to infer the hidden state sequence *Q*. The Viterbi algorithm [23] is a dynamic programming algorithm for finding the most likely hidden state sequence Q^* of the observation *O* by

$$\begin{aligned} Q^* &= \arg\max_{Q} P(Q \mid O, \lambda) \\ &= \arg\max_{Q} P(Q, O \mid \lambda). \end{aligned} \tag{6}$$

B. Selection of HMM Features

A key issue to be addressed for the HMM model is the features to be selected for the model. The features can be any subsets of the 25 joint coordinates captured by Kinect, or quantities derived from the joint coordinates (like the six KCQs defined in Section III-B). Among all the model parameters $\lambda = \{A, b_j(X), \pi_i\}$, the selection of features affects only the emission probability density $b_j(X)$. A high-dimensional feature requires more complicated GMM model and affects the accuracy of estimating $b_j(X)$ using the EM algorithm, and hence would be less desirable. Besides, the feature should reflect the difference of the emission probabilities $b_i(X)$ and $b_j(X)$ for state S_i and S_j, so that the hidden state of frames around the boundary between adjacent states/sub-actions can be correctly inferred. For the three physical therapy tasks for PD patients, the displacement d and velocity v of the related body parts are two typical features that represent the difference of patient's movements in different sub-actions. In the task SQ, the patient bends his/her legs to move the hips up and down, thus ThA represents the movement and can be used as the displacement d. In the tasks FL/BL, the patient is moving one foot back and forth so NLoS can be used as the displacement d. The velocity at time t is obtained by

$$v_t = (d_t - d_{t-1}) \cdot FR, \tag{7}$$

where FR is the frame rate (30 *fps*). Table IV shows the typical values of d and v in each state/sub-action. (Note that there may be noise added to the typical values.) For example, d and v fluctuate around zero in S_1 when the patient is standing. In S_3, the patient is maintaining the balance control at the target position so d is at the max value and v fluctuate around zero.

TABLE IV. DISPLACEMENT AND VELOCITY IN EACH SUB-ACTION

Feature	S_1	S_2	S_3	S_4	S_5
d	0	0 ~ max	max	0 ~ max	0
v	0	> 0	0	< 0	0

Although either d or v has different emission probability distributions in different states/sub-actions, using the combination of them is superior than using any single one of them as the HMM feature. The reason is as follows.

If we use only v as the HMM feature, noise in S_1 (or S_3, S_5) may have similar velocity as S_2 and S_4, thus be detected as a complete repetition (i.e., extra repetition). (Note that noise in v results from noise in d according to (7)). Fig. 5 shows an example of patient's motion data d and v when performing FL, with the x-axis showing the frame number and the y-axis showing the value of d (top) or v (bottom). If only v is used as the HMM feature, the hidden states of noise in S_1 may be detected as a complete repetition on FL (i.e., including all the states) using the HMM-M model. This problem can be solved by also considering d in the HMM feature since the displacement of such kind of noise is too small for a complete repetition.

If we use only d as the HMM feature, some complete repetitions may be included in S_2 of the next repetition or S_4 of the previous repetition (i.e., missing repetitions), especially when the time length of this repetition (i.e., number of frames) is short and the amplitude of displacement is relatively small. Fig. 6 shows an example. If only d is used as the HMM feature, the emission probability of frame t^* that is actually in S_3 of the first repetition may have $b_3(d_{t^*}) < b_2(d_{t^*})$. Thus, the first repetition will be detected as part of S_2 of the second repetition. If v is also included in the HMM feature, such kind of error can be avoided since the velocity of frame t^* is around zero and $b_3(v_{t^*}) \gg b_2(v_{t^*})$.

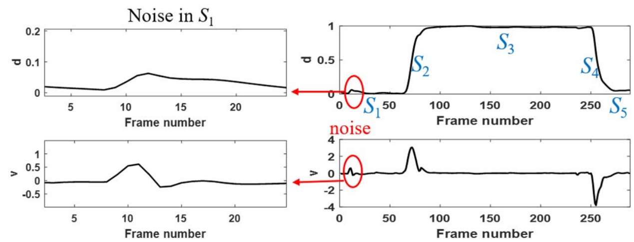

Fig. 5. Patient's motion data (d and v) when performing FL. Left: noise in S_1. Right: motion data for a complete repetition.

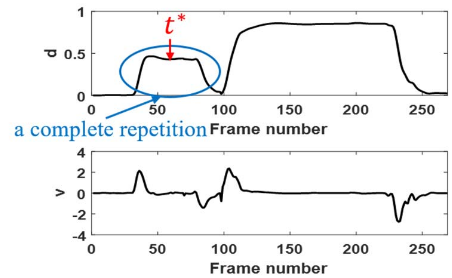

Fig. 6. Patient's motion data (d and v) when performing FL, including two repetitions.

Therefore, we propose to use the combination of d and v as the feature of the HMM model. Comparison of results using different features will be shown in Section VI-B-(1).

C. Two-Phase Human Action Understanding Algorithm

Based on the selected features (d, v) and the Viterbi algorithm [23] that is used to infer the optimal hidden state sequence from the observation, we propose the two-phase human action understanding (TPHAU) algorithm. It detects patient's repetitions in performing the task in the first phase, and segments sub-actions in each repetition in the second phase.

In the first phase, we use the HMM-M model to detect the patient's repetitions on the task. First, the HMM-M model is trained from the training samples (see Section IV-A-(1)). Considering the difference of displacement amplitude in different patients and in different repetitions, the displacement data d in each repetition are normalized into [0, 1]. For the test sample, the displacement data are normalized globally (i.e., for the entire performance including multiple repetitions) instead of being normalized in each repetition since the time interval for each repetition is unkown. Then based on the trained HMM-M model, the hidden states of the test sample can be estimated by applying the Viterbi algorithm. Patient's repetitions can be further inferred based on the state sequence. Since S_1 is the boundary between two repetitions, the starting point of each repetition (except the first one) can be estimated as the midpoint of each consecutive S_1 sequence. Fig. 7 shows an example. Four repetitions R_1, R_2, R_3, R_4 are detected from the state sequence.

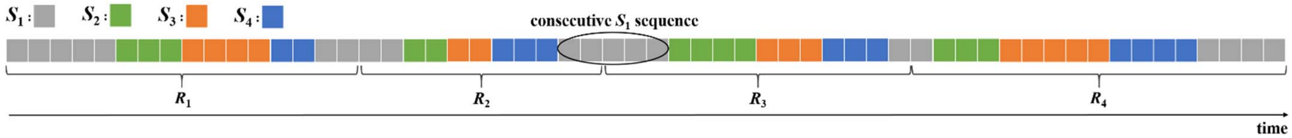

Fig. 7. State sequence obtained from the Viterbi algorithm [23]. Patient has four repetitions: R_1, R_2, R_3, R_4.

Although using both d and v as the HMM feature avoids some extra repetitions that may result from using only v as the HMM feature (see Fig. 5), it is still inevitable that noise may cause the detection of extra repetitions. There are mainly two types of extra repetitions.

1) Noise being detected as complete repetitions

Noise in S_1 may be detected as complete repetitions, as shown in Fig. 5. However, such kind of false repetition is always short in time length and the amplitude of displacement is small. Therefore, we analyze the Time Length (TL) and Amplitude of Displacement (AoD) (i.e., maximum of d) of all the repetitions in the training data. The mean value μ_{TL}, μ_{AoD} and standard deviation $\sigma_{TL}, \sigma_{AoD}$ are calculated. According to the three-sigma rule [24], a detected repetition is an outlier if

$$\left|TL-\mu_{TL}\right|>3\sigma_{TL} \text{ or } \left|AoD-\mu_{AoD}\right|>3\sigma_{AoD}. \tag{8}$$

2) Recognizing one repetition as two or more

Noise in $S_2/S_3/S_4$ may cause a complete repetition to be detected as two or more repetitions. Fig. 8 shows an example. The patient has performed two repetitions. However, the second repetition is detected as two repetitions due to noise in S_3. Such kind of extra repetitions can be eliminated by checking the value of d at the end of each repetition. In the example in Fig. 8, d at the end of the detected second repetition is too large, which means that the patient has not returned to the initial position and this repetition is not complete. Therefore, we analyze the Displacement of Endpoint (DoE) (i.e., d at the endpoint of each repetition) of all the training data. A repetition detected in the test sample is decided as extra repetition if

$$\left|DoE-\mu_{DoE}\right|>3\sigma_{DoE}. \tag{9}$$

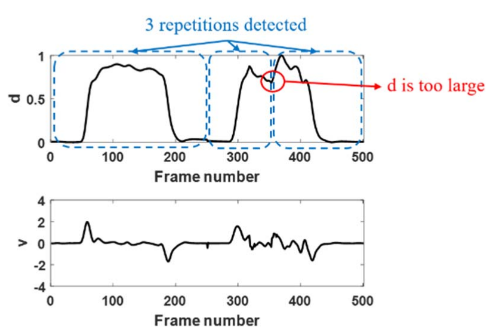

Fig. 8. Patient's motion data (d and v) with two repetitions. Three repetitions are detected.

When a detected repetition is decided as extra, it is eliminated by merging into its previous or next repetition, whichever is closer to it (i.e., the one with fewer frames of S_1 between them). After removing extra repetitions, we use a second phase to segment sub-actions in each repetition. Although the state sequence obtained from the first phase also includes information about sub-actions in each repetition, the sub-action information is not accurate due to the following reason. In the first phase, global normalization is used thus the range of d in some repetitions may be smaller than [0, 1]. Different normalization methods for the training and test data will cause the inaccuracy in state/sub-action detection. For example, in training data, d will always reach 1 in S_3 because of the repetition-based normalization. For a test sample where $d < 1$ in S_3, some frames at the beginning of S_3 may be detected as S_2. Therefore, we propose to use a second phase to enhance the accuracy in sub-action detection. First, we normalize each repetition that is detected from the first phase. Second, the HMM-S model is applied on each repetition separately. Since the HMM-S model is a left-to-right model for single repetition, it is guaranteed that no extra repetitions will be detected. The pseudo-code for the proposed TPHAU algorithm is shown in Fig. 9.

Algorithm Two-Phase Human Action Understanding (TPHAU)

Input: Two HMM models (HMM-M and HMM-S) trained from training samples
Patient's displacement sequence $D=\{d_1, d_2, \cdots d_T\}$

Output: Segmentation of sub-actions

1. Normalize D into [0, 1]
2. Calculate the velocity sequence V using (7)
3. Apply Viterbi algorithm [23] on the observation sequence $O=\{D;V\}$ using HMM-M, get the hidden state sequence Q
4. **for** each consecutive S_1 sequence in Q
5. Starting point of a repetition = midpoint of the consecutive S_1 sequence
6. **end for**
7. **for** each detected repetition
8. Calculate TL, AoD, DoE
9. **if** $\left|TL-\mu_{TL}\right|>3\sigma_{TL}$ or $\left|AoD-\mu_{AoD}\right|>3\sigma_{AoD}$ or $\left|DoE-\mu_{DoE}\right|>3\sigma_{DoE}$
10. Merge this repetition into the pervious or next repetition, whichever is closer to it
11. **end if**
12. **end for**
13. **for** each remaining repetition
14. Normalize the displacement sequence of this repetition D_{rep} into [0, 1], calculate the velocity sequence V_{rep}
15. Apply Viterbi algorithm on $O_{rep}=\{D_{rep};V_{rep}\}$ using HMM-S, get the hidden state sequence Q_{rep}
16. Segment sub-actions in this repetition based on Q_{rep}
17. **end for**

Fig. 9. Psuedo-code of the proposed TPHAU algorithm.

Note that there is no need to train HMM-S again in the second phase. Its state transition matrix A^s can be directly derived from the state transition matrix of HMM-M used in the first phase by

$$A^S = \begin{bmatrix} a_{11}^M & a_{12}^M & 0 & 0 & 0 \\ 0 & a_{22}^M & a_{23}^M & 0 & 0 \\ 0 & 0 & a_{33}^M & a_{34}^M & 0 \\ 0 & 0 & 0 & a_{44}^M & a_{41}^M \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

where a_{ij}^M is the (i, j) element of the state transition matrix of the HMM-M model. The emission probability distribution of $S_1/S_2/S_3/S_4$ in HMM-S is the same as that in HMM-M. The emission probability distribution of S_5 can directly use the distribution of S_1 since they represent the same human action.

V. Patient Error Identification

In the previous section, we propose the HMM models for patient's movements on the physical therapy tasks and the TPHAU algorithm that can detect patient's repetitions and sub-actions in each repetition. In this section, we will introduce how to identify patient's error in performing the task based on the results of sub-action detection.

For any task, the criteria used for evaluating the patient's performance have been defined by our PT co-author (see Table III). Criteria are independent of each other (i.e., whether the patient is performing correctly on one criterion is independent of his/her performance on the other criteria). Based on the human action understanding result (including the repetition detection and sub-action detection), patient's error can be identified by checking the value of the corresponding KCQs in the applied sub-actions of each criterion. For example, the criterion "keep the back knee straight" of FL applies to S_2 and S_3 (see Table III), so we just need to check the knee angle (KA) of the back leg for frames in S_2 and S_3. Patient's error in one frame e_{frame} is calculated as the difference between the patient's knee angle (KA) in this frame and the required 180 degrees. Patient's error in a repetition e_{rep} is the average of e_{frame} among all the applied frames (i.e., frames of the applied sub-actions) in this repetition. Patient's overall error on this criterion is calculated as the mean and maximum of e_{rep} for all the repetitions.

However, patient's movements may not be strictly the same as the PT-defined criteria. For example, the patient's knee angle cannot be perfectly 180 degrees even when he/she is keeping the knee straight. Thus, an error threshold is needed for each criterion. When evaluating the patient's performance, the PT gives qualitative instead of quantitative conclusions. For example, the patient's performance is either 'good or 'not good' on the criterion "keep the back knee straight". Therefore, we only need to classify the patient's error into one of the two cases. To obtain the classification model, we train a SVM model [25] from the training samples. For each training sample, the overall error on a criterion is calculated. The label y of the sample is given by the PT during the data collection process, with $y = 0$ representing negative samples (i.e., performance is not good on this criterion) and $y = 1$ representing positive samples (i.e., performance is good on this criterion). A linear SVM classification model is trained from the training data to find out the optimal decision boundary between the positive and negative samples. For any new sample, this model can be applied to decide whether the patient's performance is good or not on this criterion. Since criteria are independent of each other, a unique classification model is trained for each criterion.

VI. Experimental Results

A. Experimental Setup

In collaboration with the UC San Diego Health – La Jolla, we conducted comprehensive patient data collection at the Neurological Rehabilitation Clinic. We collected motion data from 26 PD patients (age 56~89, 15 males, 11 females) of all severity, from only mildly impaired to severe balance and mobility impairments. Each patient participated in the data collection for multiple times. In one physical therapy session, the patient performed one variation of each task (i.e., SQ, FL, BL) and his/her motion data were recorded by the Kinect sensor. The variation of each task (i.e., the difficulty level) that the patient should perform in each session was decided by the PT according to the patient's health condition. The corresponding PT evaluations (i.e., the patient's performance was good or not on each criterion) were also recorded. For each task, the movements of one patient in one session is a sample. Note that sometimes some patients were not able to perform some tasks (e.g., BL was too difficult for some patients), thus the numbers of collected samples for each task were different. We collected 57 samples for SQ, 56 samples for FL, and 45 samples for BL. Typically, patient's movements on a task includes 4 repetitions, with about 10 seconds on each repetition. The Kinect sensor captures the (x, y, z) coordinates of 25 joints per frame. With frame rate of 30 frames/second, that amounts to about 90,000 data points for each task performed by a patient in one session.

B. Human Action Understanding Results

1) Comparison of Features

As discussed in Section IV-B, using only d as the HMM feature may cause missing repetitions and using only v may lead to the detection of extra repetitions. To show the superiority of using (d, v) as the HMM feature, we use FL as an example to show the repetition detection results of using d, v, and (d, v) as the HMM feature. All the collected samples are randomly divided into a training set (including 51 samples) and a test set (including 5 samples). Three HMM-M models are trained using the training set, each with a different feature. The first phase of the proposed TPHAU algorithm is applied to each test sample. The accuracy of the repetition detection result is evaluated as follows. We check whether the starting point of each repetition is detected correctly. (Since the endpoint of each repetition is the starting point of the next repetition or the end of the entire movements, there is no need to check the endpoint). In the hidden state sequence obtained from the Viterbi algorithm (see Fig. 7 as an example), each consecutive S_1 sequence represents the start of a repetition and it is considered correct if the manually-labelled starting point of this repetition lies in this consecutive S_1 sequence. Otherwise, this detected repetition is either wrong or extra, depending on the total number of detected repetitions. When the number of detected repetitions is smaller than the number of manually-labelled repetitions, their difference represents the number of missing repetitions.

To obtain more comprehensive validation results, we repeat the training/test split process for 10 times. There are 313 repetitions in total from the 50 test samples. Table V shows the number of correct repetitions, wrong repetitions, missing repetitions and extra repetitions by using the three types of features. We can see that if only d is used as the HMM feature, there are 9 missing repetitions which is hard to recover. If we use only v, 163 extra repetitions are detected, which validates our discussion in Section IV-B. Among the three types of features, (d, v) shows the best repetition detection result as there are only 5 wrong repetitions and 8 extra repetitions. It is also demonstrated that using (d, v) as the HMM feature ensures high accuracy (97.7%) in the repetition detection.

TABLE V. NUMBER OF CORRECT, WRONG, MISSING, AND EXTRA REPETITIONS USING DIFFERENT FEATURES FOR FL

Feature	Number of repetitions			
	Correct	*Wrong*	*Missing*	*Extra*
d	194 (91.1%)	10 (4.7%)	9 (4.2%)	0
v	197 (92.5%)	16 (7.5%)	0 (0%)	163
(d, v)	208 (97.7%)	5 (2.3%)	0 (0%)	8

2) *Results of the Two-Phase Human Action Understanding Algorithm*

To validate the proposed TPHAU algorithm, experiments are conducted on SQ, FL, BL separately. (d, v) is used as the HMM feature. 85% of all the samples are randomly chosen as training set and 15% as test set. The training/test split process is repeated for 10 times. First, we want to show the effects of the proposed outlier removal method in repetition detection. Table VI shows the number of correct, wrong, missing, and extra repetitions of applying only the first phase of TPHAU (i.e., the one-phase Viterbi algorithm before the outlier removal) and the complete TPHAU algorithm (i.e., including the outlier removal method). We can see that the accuracy of repetition detection is improved, with more correct repetitions and less extra repetitions, especially for BL. Second, the results of sub-action detection within each repetition using the proposed TPHAU algorithm are shown in Fig. 10. Results using one-phase Viterbi are also shown in Fig. 10 for comparison. The accuracy evaluation metrics we use in Fig. 10 are as follows.

Accuracy Evaluation Metrics We evaluate the accuracy of each sub-action S_2/S_3/S_4 separately. (S_1 is not evaluated since it is not important for the patient's performance.) For any frame, the proposed TPHAU algorithm segments it into the sub-action RS and the manually-labelled sub-action for this frame is MS. For any sub-action $S_i (i = 2,3,4)$, all the frames are checked and each frame is classified into one of the following four categories:

- True Positive (TP): $RS = S_i$ and $MS = S_i$.
- True Negative (TN): $RS \neq S_i$ and $MS \neq S_i$.
- False Positive (FP): $RS = S_i$ and $MS \neq S_i$.
- False Negative (FN): $RS \neq S_i$ and $MS = S_i$.

Then the sensitivity ($TP/(TP+FN)$) and specificity ($TN/(TN+FP)$) of S_2/S_3/S_4 is calculated for each test sample. The average and 95% confidence interval are calculated among all the test samples. Fig. 10 shows the results, with the 95% confidence interval shown as black vertical lines. The quantitative results are in Table VII. We can see that specificity is always high (above 90%) for both one-phase Viterbi and the proposed TPHAU algorithm. For sensitivity, TPHAU enhances the sensitivity for S_3 dramatically with higher average and smaller confidence interval. For S_2 and S_4, although the average sensitivity of TPHAU is slightly lower than one-phase Viterbi (e.g., SQ), the small difference is not critical since the manual segmentation we use as the ground truth is subjective. Besides, since patients typically spend less time in S_2 and S_4, the number of true positive samples ($TP+FN$) is smaller for S_2 and S_4 than for S_3. Thus, the false negative rate ($FN/(TP+FN) = 1-\text{sensitivity}$) of S_2 and S_4 is higher and sensitivity is lower for the same number of FN frames. Therefore, the average sensitivities that TPHAU achieves in S_2 and S_4 (above 85%) are acceptable.

C. *Patient Error Identification Results*

To validate the SVM-based patient error identification method, we use the same training and test set as Section VI-B. For a test sample, the classification into positive (i.e., performance is good) or negative (i.e., performance is not good) is independent for each criterion. For example, in FL, four SVM models are trained independently from the training set for its four criteria (see Table III). Then the four models are applied on each test sample to decide whether the patient's performance is good in each criterion. The accuracy of the proposed SVM-based method is discussed both criterion-wise and patient-wise.

1) *Criterion-wise Evaluation*

For each criterion, the accuracy is calculated as the ratio of the correctly classified samples to the total number of test samples. Table VIII shows the accuracy of each criterion for the three tasks. We can see that all the accuracy values are above 85%. For some criteria, the accuracy of the SVM model is not too high because the labels (i.e., positive or negative) we use for training and testing may be subject to the PT's subjective bias.

TABLE VI. NUMBER OF CORRECT, WRONG, MISSING, AND EXTRA REPETITIONS USING DIFFERENT METHODS FOR SQ, FL, BL

Method	SQ				FL				BL			
	Correct	*Wrong*	*Missing*	*Extra*	*Correct*	*Wrong*	*Missing*	*Extra*	*Correct*	*Wrong*	*Missing*	*Extra*
One-phase Viterbi	188 (90.4%)	20 (9.6%)	0 (0%)	2	208 (97.7%)	5 (2.3%)	0 (0%)	8	196 (96.6%)	7 (3.4%)	0 (0%)	37
TPHAU	202 (97.1%)	6 (2.9%)	0 (0%)	0	209 (98.1%)	4 (1.9%)	0 (0%)	0	202 (99.5%)	1 (0.5%)	0 (0%)	2

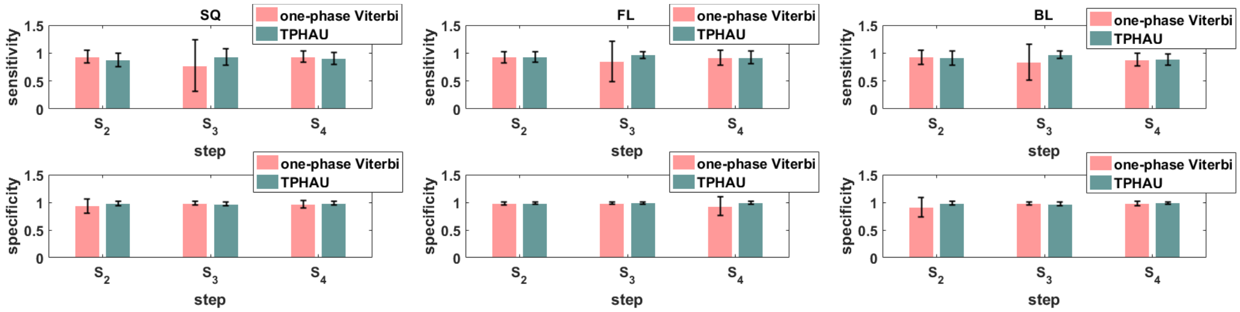

Fig. 10. Sub-action segmentation results (sensitivity and specificity of $S_2/S_3/S_4$) using different methods for SQ, FL, BL.

TABLE VII. AVERAGE SENSITIVITY AND SPECIFICITY OF $S_2/S_3/S_4$ USING DIFFERENT METHODS

	Method	SQ			FL			BL		
		S_2	S_3	S_4	S_2	S_3	S_4	S_2	S_3	S_4
Sensitivity	One-phase Viterbi	**93.4%**	77.5%	**93.7%**	92.8%	85.3%	91.3%	**92.8%**	83.2%	88.3%
	TPHAU	88.1%	**93.2%**	90.3%	**92.9%**	**96.5%**	**92.3%**	91.2%	**97.1%**	**89.0%**
Specificity	One-phase Viterbi	93.3%	**98.4%**	96.8%	97.7%	98.6%	93.2%	91.5%	**97.7%**	97.7%
	TPHAU	**97.9%**	97.2%	**98.6%**	**98.3%**	**98.8%**	**98.7%**	**98.5%**	97.2%	**99.0%**

TABLE VIII. ACCURACY OF PATIENT ERROR IDENTIFICATION MODELS FOR SQ, FL, BL

Task	Criterion	Accuracy
SQ	Sit hips back towards a chair	95.6%
	Lean forward	87.8%
FL	Keep the back knee straight	85.6%
	Keep the posture tall	97.8%
	Length of step	94.4%
	Keep the front shank vertical	91.1%
BL	Keep the back knee straight	94.3%
	Keep the trunk parallel with the back leg	88.6%
	Length of step	97.1%
	Keep the front shank vertical	87.1%

2) Patient-wise Evaluation

For each test sample, the patient-wise evaluation is represented by the number of the correctly classified criteria. Table IX shows the percentage of test samples with different numbers of correctly classified criteria. We can see that the proposed SVM-based models can predict patient's error correctly in all the criteria of the task (2 criteria for SQ, 4 criteria for FL/BL) for over 70% of the test samples. For over 95% of the test samples, the models can identify patient's error correctly except for at most one criterion (i.e., number of incorrectly classified criteria ≤ 1).

From the results of criterion-wise evaluation and patient-wise evaluation, it can be concluded that the proposed error identification models identify patient's movement error as the PT would have subjectively done with high accuracy. Therefore, the proposed models can be used to automatically evaluate patient's performance remotely, without the patient having to be physically inspected at-location by the PT, which enables effective self-training and evaluation at home.

TABLE IX. PERCENTAGE OF TEST SAMPLES WITH DIFFERENT NUMBERS OF CORRECTLY CLASSIFIED CRITERIA

Task	Number of correctly classified criteria				
	0	*1*	*2*	*3*	*4*
SQ	0%	16.7%	83.3%		
FL	0%	0%	2.2%	26.7%	71.1%
BL	0%	0%	2.9%	27.1%	70.0%

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a learning-based personalized treatment system to enable home-based training for PD patients. It captures the patient's movements at home using the Kinect sensor. Patient's movements can be understood by the proposed TPHAU algorithm. To evaluate patient's performance, we propose an SVM-based method to identify patient's error in performing the task, which emulates PT evaluation. The proposed treatment system enables low-cost and remote care for PD patients. Experiments on real patient data collected in the clinic show that the proposed methods can accurately understand patient's actions and identify patient's error. In the future, we will design a learning-based task recommendation model to enable automatic task update recommendation for PD patients. In addition, we plan to apply and enhance our approach to enable remote therapy for other kinds of disease.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under grant No. IIS-1522125.

REFERENCES

[1] Parkinson's disease statistics by Parkinson's News Today. [Online]. Available: https://parkinsonsnewstoday.com/parkinsons-disease-statistics/

[2] L. A. King, et al., "Effects of Group, Individual, and Home Exercise in Persons With Parkinson Disease: A Randomized Clinical Trial," *Journal of neurologic physical therapy: JNPT* 39.4 (2015): 204-212.

[3] B. R. Chen, et al., "A web-based system for home monitoring of patients with Parkinson's disease using wearable sensors," *IEEE Transactions on Biomedical Engineering* 58.3 (2011): 831-836.

[4] D. Pan, R. Dhall, A. Lieberman, and D. B. Petitti, "A mobile cloud-based parkinson's disease assessment system for home-based monitoring," *JMIR mHealth and uHealth* 3.1 (2015).

[5] B. Galna, et al., "Retraining function in people with Parkinson's disease using the Microsoft kinect: game design and pilot testing," *Journal of neuroengineering and rehabilitation* 11.1 (2014): 60.

[6] J. E. Pompeu, et al., "Feasibility, safety and outcomes of playing Kinect Adventures!™ for people with Parkinson's disease: a pilot study," *Physiotherapy* 100.2 (2014): 162-168.

[7] W. Wei, Y. Lu, E. Rhoden, and S. Dey, "User performance evaluation and real-time guidance in cloud-based physical therapy monitoring and guidance system," *Multimedia Tools and Applications* (2017): 1-31.

[8] Kinect. [Online]. Available: www.xbox.com/en-US/kinect

[9] J. F. Esculier, J. Vaudrin, P. Bériault, K. Gagnon, and L. E. Tremblay, "Home-based balance training programme using Wii Fit with balance board for Parkinson's disease: a pilot study," *Journal of Rehabilitation Medicine* 44.2 (2012): 144-150.

[10] J. Finkelstein, J. Wood, and E. Cha, "Impact of physical telerehabilitation on functional outcomes in seniors with mobility limitations," in Proc. of *the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC 2012), San Diego, CA, USA, Aug. 2012.

[11] C. Jeong and J. Finkelstein, "Introducing Telerehabilitation in Patients with Multiple Sclerosis with Significant Mobility Disability: Pilot Feasibility Study," in Proc. of the 2015 International Conference on Healthcare Informatics (ICHI 2015), Dallas, TX, USA, Oct. 2015.

[12] B. Galna, et al., "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease," *Gait & posture* 39.4 (2014): 1062-1068.

[13] T. Y. Lin, C. H. Hsieh, and J. D. Lee, "A kinect-based system for physical rehabilitation: Utilizing tai chi exercises to improve movement disorders in patients with balance ability," in Proc. of *the 2013 7th Asia Modelling Symposum* (AMS 2013), Hong Kong, China, Jul. 2013.

[14] L. Xia, C. C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in Proc. of *the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW 2012), Providence, RI, USA, Jun. 2012.

[15] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in Proc. of *the 2012 IEEE International Conference on Robotics and Automation* (ICRA 2012), Saint Paul, MN, USA, Jun. 2012.

[16] H. Pirsiavash, and D. Ramanan. "Parsing videos of actions with segmental grammars," in Proc. of *the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2014), Columbus, OH, USA, Jun. 2014.

[17] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in Proc. of *the 2015 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2015), Boston, MA, USA, Jun. 2015.

[18] L. A. King, F. B. Horak, "Delaying mobility disability in people with Parkinson disease using a sensorimotor agility exercise program," *Physical Therapy* 89.4 (2009): 384-393.

[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE* 77.2 (1989): 257-286.

[20] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics* 41.1 (1970): 164-171.

[21] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal* 62.4 (1983): 1035-1074.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)* (1977): 1-38.

[23] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE* 61.3 (1973): 268-278.

[24] F. Pukelsheim, "The three sigma rule," *The American Statistician* 48.2 (1994): 88-91.

[25] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery* 2.2 (1998): 121-167.