Title

Using Sensors and AI to Enable On-Demand Virtual Physical Therapist and Balance Evaluation at Home

Author

Wei, Wenchuan

Publication Date

2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Using Sensors and AI to Enable On-Demand Virtual Physical Therapist and Balance Evaluation at Home

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Computer Engeneering)

by

Wenchuan Wei

Committee in charge:

Professor Sujit Dey, Chair
Professor Todd Coleman
Professor Pamela Cosman
Professor Truong Nguyen
Professor Erik Viirre

2020

The dissertation of Wenchuan Wei is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Sujit Dey for his continuous support of my Ph.D. study and research. Under his careful supervision and guidance, I have learned a lot in scientific research and improved my writing and presentation skills in a systematic way.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Pamela Cosman, Prof. Truong Nguyen, Prof. Todd Coleman, and Prof. Erik Viirre, for their insightful comments and suggestions. I would also like to thank the physical therapist collaborators who worked closely with me on this project: Mr. Carter McElroy, Mr. Eric Rhoden, and Ms. Catherine Printz. Their expertise in the field of physical therapy helped me a lot in conducting the study. I am also grateful to every member of the MESDAT lab at UCSD for making my experience in the lab exciting and fun.

Last but not least, I would like to express my deepest gratitude to my family and friends. This dissertation would not have been possible without their warm love, continued patience, and endless support.

Chapter 2, in part, is from the material as it appears in proceedings of ACM conference on Wireless Health 2015. Wenchuan Wei; Yao Lu; Catherine D. Printz; Sujit Dey. and in Multimedia Tools and Applications 2017. Wenchuan Wei; Yao Lu; Eric Rhoden; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

Chapter 3, in part, is from the material as it appears in proceedings of IEEE International Conference on Healthcare Informatics 2018. Wenchuan Wei; Carter McElroy; Sujit Dey. and in IEEE Transactions on Neural Systems & Rehabilitation Engineering 2019. Wenchuan Wei; Carter McElroy; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

Chapter 4, in part, is from the material as it appears in proceedings of IEEE International Conference on Healthcare Informatics 2019, Wenchuan Wei; Sujit Dey. and in IEEE Access 2020.

Wenchuan Wei; Carter McElroy; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

VITA

2014	Bachelor of Engineering, Tsinghua University
2014-2020	Research Assistant, University of California San Diego
2017	Master of Science, University of California San Diego
2020	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

W. Wei, C. McElroy, S. Dey, "Using Sensors and Deep Learning to Enable On-Demand Balance Evaluation," *IEEE Access*, vol. 8, pp. 99889-99899, May 2020.

W. Wei, C. McElroy, S. Dey, "Towards On-Demand Virtual Physical Therapist: Machine Learning-Based Patient Action Understanding, Assessment and Task Recommendation," *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, vol. 27, no. 9, pp. 1824-1835, September 2019.

W. Wei, and S. Dey, "Center of Mass Estimation for Balance Evaluation Using Convolutional Neural Networks," *Proceedings of the Seventh IEEE International Conference on Healthcare Informatics*, Xi'an, China, June 2019, pp. 1-7.

W. Wei, C. McElroy, S. Dey, "Human Action Understanding and Movement Error Identification for the Treatment of Patients with Parkinson's Disease," *Proceedings of the Sixth IEEE International Conference on Healthcare Informatics*, New York City, June 2018, pp. 180-190.

W. Wei, Y. Lu, E. Rhoden, S. Dey, "User Performance Evaluation and Real-Time Guidance in Cloud-Based Physical Therapy Monitoring and Guidance System," *Multimedia Tools and Applications*, November 2017, pp. 1-31.

W. Wei, Y. Lu, C. Printz, S. Dey, "Motion Data Alignment and Real-Time Guidance in Cloud-Based Virtual Training System," *Proceedings of ACM conference on Wireless Health*, Bethesda, MD, October 2015, pp. 1-8.

ABSTRACT OF THE DISSERTATION

Using Sensors and AI to Enable On-Demand Virtual Physical Therapist and Balance Evaluation at Home

by

Wenchuan Wei

Doctor of Philosophy in Electrical Engineering (Computer Engeneering)

University of California San Diego, 2020

Professor Sujit Dey, Chair

The effectiveness of traditional physical therapy may be limited by the sparsity of time a patient can spend with the physical therapist (PT) and the inherent difficulty of self-training given the paper/figure/video instructions provided to the patient with no way to monitor and ensure compliance with the instructions. In this dissertation, we propose a virtual PT system using sensors and AI to enable on-demand physical therapy training and balance evaluation at home. This work can be divided into three stages. Firstly, we have developed a cloud-based monitoring and guidance system for home-based physical therapy training. We use a motion capture sensor to track the patient's performance and develop algorithms to address the latency problems in evaluating the patient's performance. Different types of guidance have been designed to help the patient improve the performance. The proposed system is a generalized model that can be applied to many types of diseases, as well as fitness training, ergonomics training, etc. Secondly, we focus on patients with Parkinson's disease (PD) and propose an action understanding, assessment, and task recommendation system. The proposed system is able to understand the patient's movements and identify the movement error. In addition, the proposed system provides personalized task recommendations for the patients. The task recommendations can be fully automated, or if desired, the system may require remote supervision and approval by the PT. Thirdly, we propose an automated balance evaluation system using multiple sensors to enable on-demand balance evaluation at home. The proposed balance evaluation model is able to provide a quantified balance level that is consistent with the human PT's assessments in traditional balance evaluation tests. To train and validate the proposed systems, we have collected real patient data from the clinic. Experimental results show high accuracy of the proposed systems. By using inexpensive sensors and AI, the proposed virtual PT and balance evaluation system has the potential of enabling on-demand virtual care and significantly reducing cost for both patients and care providers.

Chapter 1

Introduction

1.1 Background and System Architecture

In recent years, the emergence of various medical sensors and monitoring devices has led to the widespread development of smart healthcare which can provide cheaper, faster, and more effective monitoring and treatment for patients [1, 2, 3, 4, 5]. As a widely used type of rehabilitation in the treatment of many diseases, physical therapy is a promising field in smart healthcare applications. Traditional physical therapy involving regular visits to the physical therapist (PT) can be expensive and even unaffordable for many patients. Even if the patients are instructed in therapy sessions, they need to practice at home by following paper, figure, or video instructions, which cannot track the patient's performance and provide effective feedback.

To address this problem, virtual training systems based on rendering technologies and motion capture sensors have been developed [7, 8, 39, 40, 41, 42, 47]. In the meantime, the use of mobile devices has become pervasive – in 2019, there are 2.7 billion smartphone users and 1.35 billion tablet users across the world and 57% of all digital media usage comes from mobile apps [9]. Moreover, cloud computing has started being used as an alternative approach for mobile health applications [10], computer games [11], etc., to make up the inherent hardware constraint of mobile devices in memory, graphics processing and power supply when running heavy multimedia and security algorithms. In cloud-based mobile applications, all the data and videos are processed and rendered on the cloud, which makes it superior to local processing on desktop computers for its portability across multiple platforms. Thus, this solution can enable the users to use the system at home or away, e.g. at hotels while traveling, making it more flexible and usable. In addition, artificial intelligence (AI) has also been increasingly used in healthcare. The global AI market in healthcare is expected to grow at a compound annual growth rate of 43.5% to reach USD 27.60 billion by 2025 [12].

Therefore, combining the above technology trends, we propose a cloud-based training, monitoring and guidance system for patients who need physical therapy. The proposed system integrates expertise in seemingly disparate disciplines - computer vision, rendering technology, cloud computing, machine learning, and human factors - towards an integrated solution that holds great promise to transform physical therapy through a quantitative process that can be done at home or at the workplace. The user can use this system on a mobile device and receive real-time evaluation and guidance on an on-demand basis. The proposed system has the following features.

Cloud-based data storage and processing. All the patient data are stored and processed on the cloud. The training instructions and videos are rendered on the cloud and sent to the user's mobile device.

Avatar-based instructions. A pre-recorded avatar instructor is rendered to instruct the user and provide guidance. The user can also see his/her own avatar.

Real-time guidance. The proposed system provides real-time guidance to the patient based on his/her performance. We have also explored the effectiveness of different types of guidance.

Patient monitoring. We use a motion capture sensor and a pressure sensor to track the patient's performance.

Performance evaluation and task recommendation. We have developed algorithms to evaluate the patient's performance on the training tasks and provide task recommendations.

Balance evaluation. We have proposed a system using multiple sensors to evaluate the patient's dynamic balance during a simple gait initiation exercise.

Figure 1.1 shows the architecture of the proposed system. During offline data collection sessions, we have collected real patient data and trained a virtual PT model using machine/deep learning methods. During a live home session, the patient can use a mobile device to access the virtual PT model remotely. Avatar-based instructions and guidance are rendered on the cloud and sent to the patient's device. The patient's data are tracked by multiple sensors and sent back to the cloud for analysis. Based on the patient's performance, the virtual PT model will provide accurate performance evaluation, guidance, task recommendations, and balance evaluation. The PT can supervise the entire process remotely. In the following chapters, we will introduce each component of the proposed system in more details. The proposed system has the advantages of providing accurate, on-demand and personalized care. It has the potential of significantly reducing clinic visit requirements while offering continuous care, thereby reducing cost and expanding care for economically disadvantaged and rural patient populations.

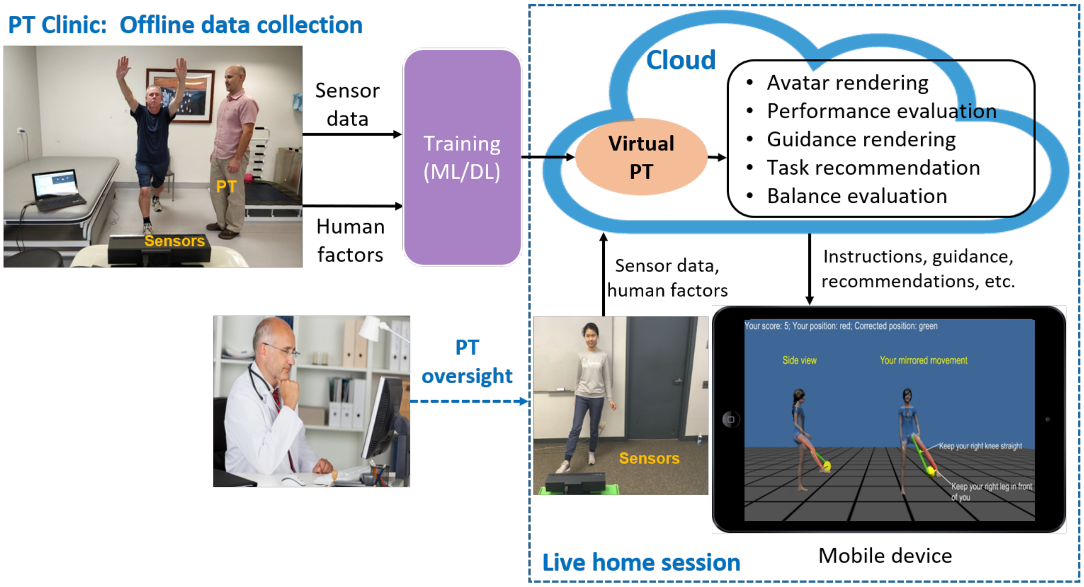


Figure 1.1: Architecture of the proposed virtual PT and balance evaluation system.

Note that in this dissertation, we use the terms “user” and “patient” interchangeably. The rest of this dissertation is organized as follows: In Chapter 2, we propose a cloud-based physical therapy monitoring and guidance system. The system is a generalized model that can be used by any type of patients who need physical therapy. In Chapter 3, we focus on the performance evaluation and task recommendations for patients with PD. In Chapter 4, we propose an automated balance evaluation system using multiple sensors to enable on-demand balance evaluation for patients with balance problems. Finally, Chapter 5 will conclude this dissertation and discuss future work in this area.

1.2 Related Work

In this section, we will introduce the related work on automated training systems for physical therapy. In the following chapters, we will introduce the related work on each component of the proposed virtual PT and balance evaluation system (shown in Fig. 1.1) in more details.

With the development of motion capture sensors, more and more sensor-based automated training systems have been developed to improve the effectiveness of home-based physical therapy training. Mirelman et al. used the marker-based optical motion capture system Vicon and proved its effectiveness in gait analysis on subjects with hemiparesis caused by stroke [8]. Ananthanarayan et al. developed a wearable electronic device called Pt Viz for knee rehabilitation [15]. However, wearable sensors attached on the body may cause extra burden to the patients. Therefore, camera-based sensors were considered more convenient in monitoring the patients’ movements in physical therapy. Microsoft Kinect [6] is proved of high accuracy and more convenient in detecting the human skeleton compared with wearable devices [16]. Lange et al. developed a game-based rehabilitation system using Kinect for balance training [17]. Chang et al. used Kinect to track arm movements to help young adults with motor disabilities [18]. In our proposed system, Kinect is used to track physical therapy tasks for its efficiency in full-body and limb tracking, as well as being readily available, easy to setup, and low-cost. However, the game-based systems [17, 18] used Kinect to motivate the patients and cannot enable careful monitoring of desired patient performance and subsequent task recommendations like a human PT does. In comparison, our proposed system is superior to the above Kinect-based systems for its high accuracy and reliability in user performance evaluation and guidance design, while will be discussed in the following chapters.

Chapter 2

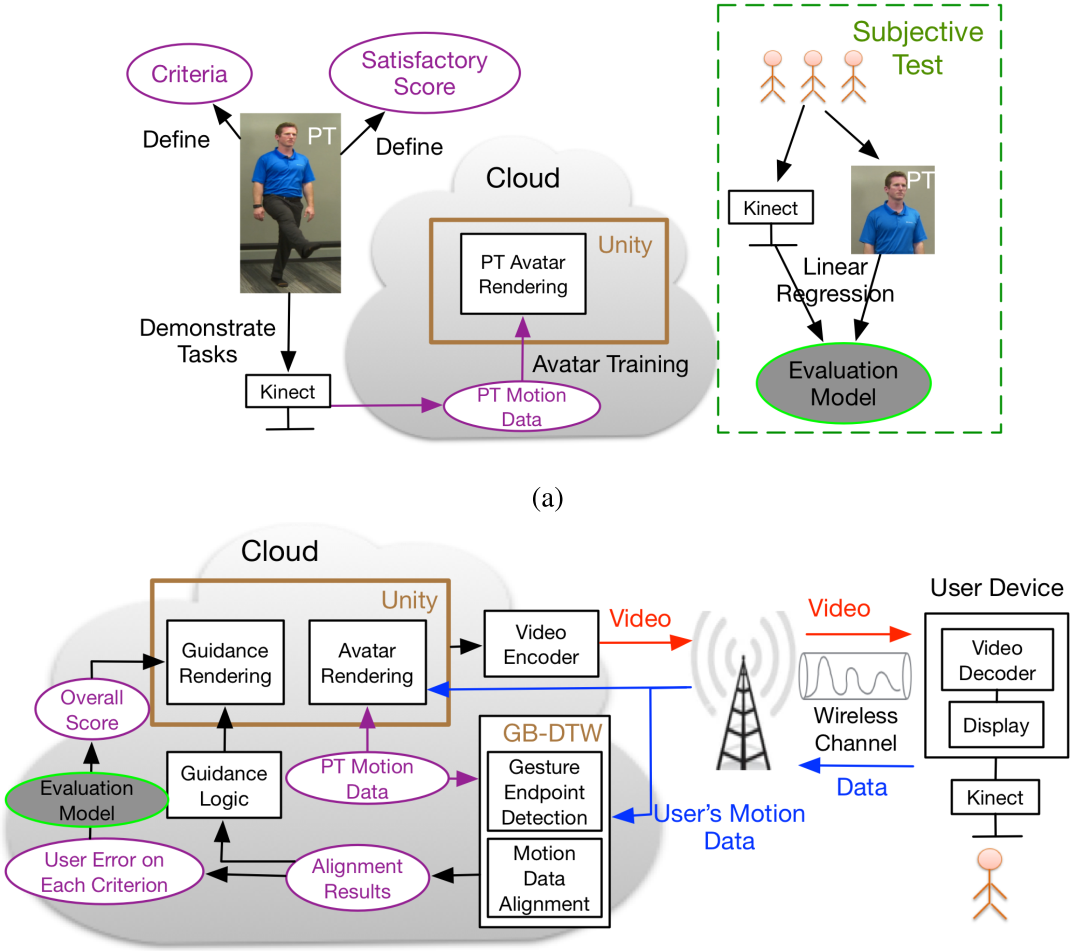
Cloud-Based Physical Therapy Monitoring and Guidance System

2.1 Introduction

In this chapter, we combine 1) rendering technology, 2) motion capture based on Microsoft Kinect [6] and 3) cloud computing for mobile devices to propose a cloud-based real- time physical therapy instruction, monitoring and guidance system. The proposed system enables a user to be trained by following a pre-recorded avatar instructor, monitors and quantifiably measures user performance, and provides real-time textual and visual guidance on his/her mobile device as needed to improve the user's performance.

The architecture of the proposed cloud-based physical therapy monitoring and guidance system is shown in Figure 2.1. Note that the physical therapy tasks discussed in this dissertation are movement based tasks. Figure 2.1(a) shows the offline session, in which a PT defines the criteria and satisfactory score for a task, and also demonstrates the task, with his/her motion data captured by the Kinect sensor and his/her avatar recorded and trained on a game development platform Unity [13]. For each task, an evaluation model is trained from a subjective test, which is used to evaluate the user's performance on this task. Figure 2.1(b) shows the online home session. A training video is transmitted through a wireless network to the user device. The user watches the training video and tries to follow the task. Simultaneously, his/her movements are captured by Kinect and uploaded to the cloud. On the cloud, the proposed Gesture-Based Dynamic Time Warping algorithm segments the user's motion sequence into gestures and aligns the motion data of the PT and user in real time. The user's accuracy is determined by transforming the user's errors into an overall score using the evaluation model obtained from the offline session. The alignment results are processed by a guidance logic. The user can progress to the next task if and when his/her accuracy reaches a satisfactory score, otherwise a guidance video is rendered and transmitted to the user device to help the user calibrate his/her movements.

The proposed system has the ability to more effectively and efficiently train people for different types of tasks, like knee rehabilitation, shoulder stretches, etc. Although other avatar-based training systems exist, our system provides real-time guidance rather than just providing scores. This feature allows the system to cater to the abilities of the user and to react to the user's performance by demonstrating the necessary adjustments to establish optimal conditions. In essence, our system is dynamic, allowing every user experience to be distinct. Moreover, together with the offline step of capturing and training an avatar for the PT tasks customized to a particular user, the proposed system enables personalized physical therapy training. Although the platform has the advantages as mentioned above, human reaction delay (delay by user to follow instructions) and wireless network delay (which may delay when the cloud rendered avatar video reaches the user device) may cause challenges for correctly calculating the accuracy of the user's movements compared to the PT avatar's movements. In particular, the delay may cause the two motion sequences to be misaligned with each other and make it difficult to judge whether the user is following the PT avatar correctly. Therefore, we apply Dynamic Time Warping (DTW) algorithm to address the problem of motion data misalignment. Considering the fact that DTW can only be applied after the user finishes the whole task, we further propose the Gesture-Based Dynamic Time



(a)

(b)

Figure 2.1: Architecture of cloud-based physical therapy monitoring and guidance system. (a) Offline session. (b) User home session.

Warping algorithm to segment the whole user motion sequence into gestures to enable real-time evaluation and guidance for the user. To evaluate the user's performance correctly, an evaluation model is trained by collecting data from subjective test and based on the professional advice of our PT collaborator. To help the user improve accuracy, we design visual/textual/combined guidance and conduct subjective test to validate their effectiveness. We have implemented the proposed algorithms in a prototype avatar based real-time guidance system and conducted experiments using wireless network profiles and on a real cloud environment. Experimental results show the performance advantage of our proposed method over other alignment methods, as well as the feasibility and effectiveness of our proposed cloud-based physical therapy monitoring and guidance system.

A preliminary version of this work has been reported in [14]. Compared with [14], we have developed a new real-time monitoring and guidance system in this chapter using Unity [13], which enables more effective avatar modeling, user performance tracking, and guidance design and delivery. The motion data are extended from one dimension to multi dimensions. In user performance evaluation, we present a new Gesture-Based Dynamic Time Warping algorithm which significantly enhances the accuracy of gesture segmentation and reduces segmentation delay, compared to the algorithm we presented in [14]. (In the rest of this chapter, we use GB-DTW0 to refer to the algorithm proposed in [14] and GB-DTW-A to refer to the new algorithm proposed in this chapter where "A" means more accurate segmentation.) Experimental results are provided to demonstrate the superior performance of the new GB-DTW-A algorithm. Furthermore, the user performance evaluation model is completely redesigned based on a procedure involving subjective testing. A new guidance system is designed which can provide more intuitive and detailed guidance. Effectiveness of the proposed real-time guidance, not discussed in [14], is validated with a new subjective study.

The rest of the chapter is organized as follows: Section 2.2 reviews related work on user performance evaluation techniques and guidance systems in physical therapy. In Section 2.3, we introduce the construction of motion data and the data misalignment problem. Section 2.4 proposes the data alignment approach and the evaluation model for the user's performance, as well as the guidance design in the proposed system. Section 2.5 presents the experimental results of motion data alignment and performance evaluation using real network profiles and on a real cloud environment, and also validates the effectiveness of guidance. Section 2.6 concludes this chapter.

2.2 Related Work

2.2.1 User Performance Evaluation

In physical therapy, patients' movements need to be carefully controlled due to their reduced mobility and the potential for re-injury. Therefore, user performance evaluation is an important part in these automatic training systems to remind patients of any incorrect motion. To evaluate the user's performance, authors in [19] propose to compare the skeletons of the trainee and the trainer tracked by Kinect sensor. First, skeleton of the trainee is scaled by resizing each bone to match the size of the corresponding bone of the trainer. Then the two skeletons are aligned by aligning the hips which are considered to be the hierarchical center of the skeleton. Finally, the trainee's performance can be evaluated by calculating the Euclidean distance between the trainee's and trainer's joints.

However, the assumption of this approach is that the trainee follows the trainer timely since they use a window of 0.5 s for any target frame to search for the best matching posture. For some challenging tasks, it might be difficulty for the user, especially for patients with injuries, to catch up with the trainer's movements. In this case, motion data of the trainer and the trainee are mismatched and the best matching posture cannot be found within the 0.5 s window.

To address the misalignment problem, authors in [20] propose to use Maximum Cross Correlation (MCC) to calculate the time shift between the standard/expected motion sequence and the user's motion sequence. Then by shifting the user's motion sequence by the estimated time shift, the two sequences are aligned and their similarity can be calculated. However, this approach assumes uniform delay during the user's movements and cannot address the problem of motion data distortion, which will be discussed in Section 2.3.2.

In [21], a training system based on wearable sensor use DTW to detect and identify correct and incorrect executions in an exercise. It is aimed at finding the best match of the user's execution among some correct and incorrect templates to judge the user's performance and give the error type if any. However, error templates can hardly cover all the mistakes patients may make, and computation increases with more templates. Besides, it can only be applied offline when the entire user motion sequence is obtained. In comparison, the proposed system does not need any pre-recorded error template. Besides, the proposed GB-DTW-A algorithm enables real-time evaluation and guidance for the user.

2.2.2 Guidance Design

To help the user improve performance, many types of guidance system have been designed. OctoPocus [22] and ShadowGuides [23] teach user gestures and movements on touch screens. LightGuide [24] projects guidance hints directly on a user's body to guide the user in completing the desired motion. In [15], wearable sensor made of lighted fabric visualizes the correct knee angle for knee rehabilitation exercises. BASE [25] based on kinematic sensor designed for older adults displays colored markers overlaid on the body to show the user's position and target position. In [19], an augmented reality mirror and colored circles/lines overlaid on the user's body are used to instruct the user and label incorrect movements. In [26], an on-screen "Wedge" visualization overlaid on top of the user's body shows the plane and range of movement, joint positions and angles, and extent of movement.

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

2.3 Motion Data Construction and Data Misalignment Problem

In the proposed system, Kinect captures 25 joints with 3-D coordinates for each joint [27]. However, only some parts of these joints are deemed important for a specific task. In this section, we will introduce how to construct the motion data for a task and the motion data misalignment problem in the system.

2.3.1 Motion Data Construction

For a given task, our PT collaborator defines several criteria and the tolerable error threshold for each criterion, which need to be translated into motion features. Motion features are quantities that are derived from the joint coordinates captured by Kinect, such as joint positions, joint angles, joint velocity, etc. For example, in the shoulder abduction task, arm height or shoulder angle (i.e., angle between the arm and the vertical direction) can be a motion feature which indicates whether the user raises the arm highly enough. Considering the difference in body size, we use normalized features, like angles, to build the motion data. The first three columns in Table 2.1 show the examples of some criteria defined by our PT collaborator, the corresponding motion features, and the tolerable error threshold for a leg lift task.

Table 2.1: Examples of task criteria and motion features of leg lift task.

Criterion	**Motion Features**	**Error Threshold**	**Feature Type**
"Lift right leg to the required height"	Angle between right leg and vertical direction: 60°	±5°	Time-varying
"Keep right knee straight"	Angle between right thigh and right shank: 180°	±10°	Constraint
"Keep right leg in front of the body"	Angle between right leg and the patient's right direction: 90°	±10°	Constraint

Moreover, there are two types of features: time-varying features and constraint features. In a task, the patient is instructed to move some parts of his/her body, and keep some other parts stationary in the meantime. Time-varying features are features which represent the body's movements in this task. Constraint features represent the other body parts which should be kept stationary during the task. The fourth column in Table 2.1 shows the corresponding feature type of each criterion in the leg lift task. For a given task, the PT defines N_v time-varying features and N_c constraint features. Time-varying motion data Fv for this task can be obtained by combining all the time-varying motion features of each frame.

$$F^v = \begin{bmatrix} f^v_{1,1} & f^v_{1,2} & \cdots & f^v_{1,N_v} \\ f^v_{2,1} & f^v_{2,2} & \cdots & f^v_{2,N_v} \\ \vdots & \vdots & \ddots & \vdots \\ f^v_{T,1} & f^v_{T,2} & \cdots & f^v_{T,N_v} \end{bmatrix} \tag{2.1}$$

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

$$F^c = \begin{bmatrix} f^v_{1,1} & f^v_{1,2} & \cdots & f^v_{1,N_c} \\ f^v_{2,1} & f^v_{2,2} & \cdots & f^v_{2,N_c} \\ \vdots & \vdots & \ddots & \vdots \\ f^v_{T,1} & f^v_{T,2} & \cdots & f^v_{T,N_c} \end{bmatrix} \tag{2.2}$$

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

2.3.2 Motion Data Misalignment Problem

Given the motion data of the PT and the user, we calculate the similarity of the two sequences to evaluate the performance of the user. However, comparing the two sequences directly is unreliable due to the potential data misalignment caused by delay. There are mainly two kinds of delay in the system: 1) human reaction delay, which means that it may take the user some time to react to the demonstration task before following it, 2) network delay, which results from the wireless network when transmitting the training video from the cloud to the user device. Human reaction delay and network delay cause two types of motion data misalignment problem: time shift and data distortion. In the rest of this section, we will discuss these two types of data misalignment problem, and discuss the problems the existing technique MCC [20] has in addressing the misalignment between the two sequences. 1) Time Shift Delay When human reaction delay and network delay are uniform in a training task, there is only time shift between the PT's and the user's motion data. In this case MCC can be used to estimate the time shift and align the two sequences. For two discrete-time signals f and g, their cross correlation Rf,g(n) is defined by

$$F_{f,g}(n) = \sum_{m=-\infty}^{\infty} f^*(m)g(m+n), \tag{2.3}$$

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

$$\tau = \arg\max_{n}\{R_{f,g}(n)\}. \tag{2.4}$$

For those tasks including multiple separate gestures, the time shift might be different for these gestures and need to be calculated separately. Here we define a gesture as a subsequence that represents an independent subtask, e.g., one- time shoulder abduction and adduction. Gestures in a training task are segmented manually by our PT collaborator. Figure 2.2 shows a simple example of the PT and user's motion data in a task of three gestures. For each gesture, the user follows the PT avatar to perform shoulder abduction and adduction. Figure 2.2(b) shows the angle between the left arm and the vertical direction as an example of the motion feature. Suppose that the user performs each gesture with delay τ_1, τ_2 and τ_3 ($\tau_1 \neq \tau_2 \neq \tau_3$), they can be estimated using MCC and the two sequences can be aligned by shifting each gesture by the corresponding estimated delay.

2) Motion Data Distortion

In many cases, human reaction delay and network delay may not be uniform. The user may not be able to follow the task timely or perform some incorrect motion when the task is difficult for him/her. For example, when following a task of 2 seconds, it takes a user 1s to react to the instructions and another 1 s to complete the task since he realizes that he is behind. In this case the user's reaction delay is not uniform (delay = 1 s when $t \leq 1$ s, delay < 1 s when 1 s < t < 2 s, and delay = 0 when t = 2 s). Besides, the user's valid motion sequence (1 s) is shorter than the PT's (2 s), so shifting one sequence by the estimated delay cannot effectively align them. Network delay may also be not uniform due to many factors, such as varying bandwidth and network load. Although some response time management techniques have been developed [28], the network delay in cloud mobile applications cannot be eliminated. Therefore, under the influence of fluctuating network delay or when the user is following some difficult tasks, the user's motion data might be distorted compared with the PT's. Figure 2.3 shows the motion data of the same task as Figure. 2.2, but with both time shift delay and motion data distortion. In this case, using MCC to shift the user's sequence by an estimated delay is unreliable. To calculate the similarity between the two sequences effectively, we need to find an optimal way to align them.

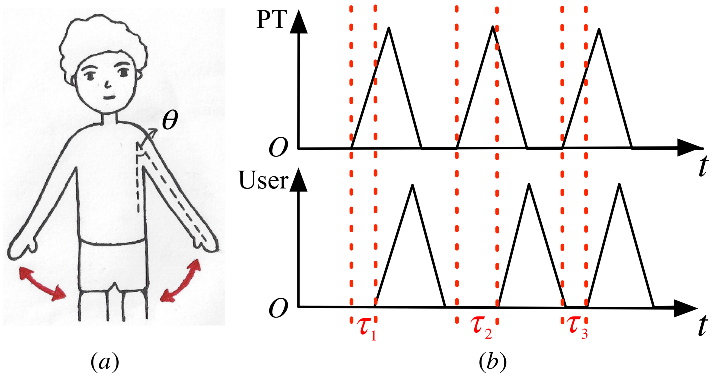


Figure 2.2: (a) Shoulder abduction and adduction. (b) Motion data (i.e., angle between left arm and the vertical direction) of the PT and user for three gestures with only time shift delay. Delay for each gesture is τ_1, τ_2, τ_3.

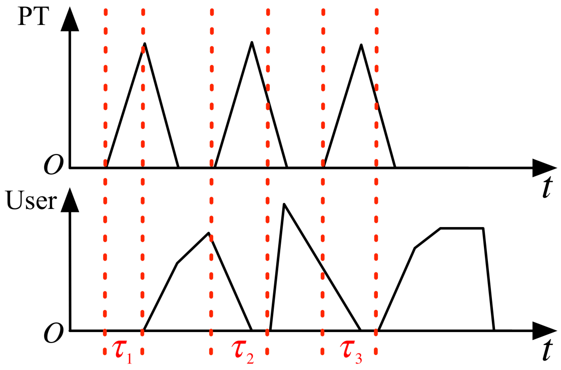


Figure 2.3: Motion data (i.e., angle between left arm and vertical direction) of the PT and the user with both time shift delay and motion data distortion.

2.4 Motion Data Alignment and User Performance Evaluation

To solve the data misalignment problem and evaluate the user's performance correctly, we propose a DTW-based data alignment and evaluation method. Section 2.4.1 introduces the principle of classical DTW and its use in the proposed system. Section 2.4.2 proposes the GB-DTW-A algorithm which segments user gestures so that data alignment can be done in real time based on each gesture, and introduces the enhancements of GB-DTW-A compared with the original GB- DTW0 algorithm [14]. In Section 2.4.3, we discuss how to evaluate the user's performance according to the alignment results of GB-DTW-A. Finally, Section 2.4.4 introduces visual and textual guidance in the proposed system and discusses how to provide effective guidance for the user.

2.4.1 Dynamic Time Warping

DTW is a dynamic programming algorithm that is widely used in speech processing [29]. It measures the similarity between two sequences $A = \{a_1, a_2, \cdots, a_m\}$ and $B = \{b_1, b_2, \cdots, b_n\}$ by calculating their minimum distance. To calculate the minimum distance, an $m \times n$ distance matrix D is defined where $D(i,j)$ is the Euclidean distance between a_i and b_j.

$$D(i,j) = \|a_i - b_j\| \tag{2.5}$$

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

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

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

$$S(i+j) = D(i,j) + min\begin{cases} S(i-1,j-1) \\ \quad S(i,j-1) \\ \quad S(i-1,j) \end{cases} \tag{2.6}$$

and $S(m,n)$ is defined as the DTW distance between A and B [30]; smaller DTW distance indicates that the two sequences are more similar. The optimal warping path can be found by backtracking from (m,n) to $(1,1)$ and this path indicates the best way to align the two sequences. Time complexity of the DTW method is $O(mn)$. Figure 2.4(a) shows an example of two sequences A and B. The purple marked elements construct a path from $(1,1)$ to (m,n) on which the accumulative distance is minimized. It is the optimal warping path between A and B. Figure 2.4(b) shows the corresponding alignment given by the optimal path in Figure 2.4(a). For example, a1 is aligned with b_1, a_2 and a_3 are aligned with b_2.

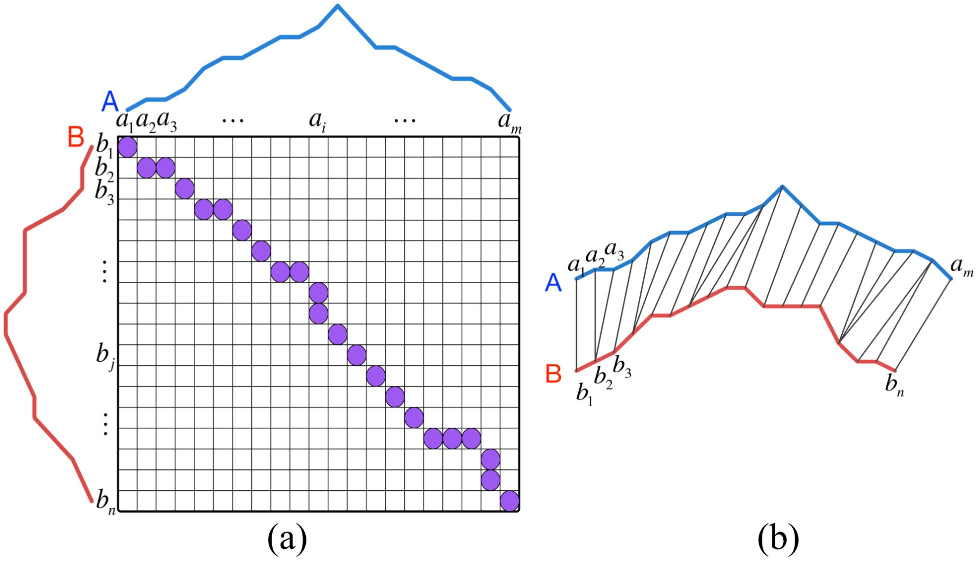


Figure 2.4: (a) Warping path of DTW on sequence A and B. (b) Alignment results between A and B.

In the proposed system, DTW can be used to find out the optimal alignment between the PT's and user's movements. As mentioned in Section 2.3.1, there are two types of motion data: time-varying motion data Fv and constraint motion data Fc. For time-varying motion data F_v, delay problems mentioned in Section 2.3.2 may cause the data to be misaligned with each other. Therefore, DTW can be applied to the PT's and user's time-varying motion data to find out an optimal warping path $P = \{p_1, p_2, \cdots, p_q\}$, where $p_k = (x_k, y_k)$ indicates that the user's performance in frame y_k matches PT's movement in frame x_k. Constraint motion data vs. time are horizontal lines (e.g., the knee angle vs. time is a horizontal line at 180 degrees for criterion "keep knee straight" in Table 2.1 and DTW cannot be used to align them. Constraint motion data are aligned using the DTW alignment results of time- varying motion data. Consequently, based on the alignment results, the user's performance can be evaluated by comparing his/her movements with the PT's demonstration movements.

2.4.2 Real-Time Gesture Segmentation Based on DTW

Although DTW is an effective way to find out the optimal alignment between the PT and user's motion sequences, it works only after the two motion sequences are obtained, that is, after the user finishes the entire task. In the proposed system, we would like to provide real-time evaluation for the user after he/she finishes each gesture, thus real-time gesture segmentation is needed during the user's performance. There has been numerous research in the field of gesture segmentation, including methods based on machine learning, signal processing [31], [32], etc. In this work, since DTW can be used to align the motion sequences, we further propose a variant of DTW called GB-DTW-A so that gesture segmentation can be implemented in the process of DTW. We next present the details of GB-DTW-A. For a given task, gestures in the PT's motion sequence have been pre-defined and segmented, which will be used as the ground truth to segment user's gestures. Suppose that $A_1 = \{a_1, a_2, \cdots, a_{m_1}\}$ is defined as the first gesture in the PT's motion sequence A. Then we would like to use DTW to find a subsequence $B_1 = \{b_1, b_2, \cdots, b_k\}(2 \leq k \leq n)$ of the user's motion data B that matches the PT's gesture A_1 best. Since the DTW distance $S(m_1, k)$ represents the similarity between A_1 and B_1, the optimal endpoint n_1 of the user's gesture should be the position with the minimum DTW distance.

$$n_1 = \underset{2 \leq k \leq n}{\arg\min}\{S_{m_1,k}\}. \tag{2.7}$$

In [30], the Subsequence DTW algorithm searches the entire user sequence B to find out the global optimum n_1. However, it works only after the user completes the entire task, which means that it is not real-time. Here we propose a new approach to estimate the global optimum by testing each local optimum. Firstly, we define a normalized distance function $T(k) = S(m_1, k)/(\sum_{i=1}^{m_1} a_i)$, where $\sum_{i=1}^{m_1} a_i$ is the sum of PT's motion data on this gesture. Then $T(k)$ can be used as a uniform similarity metric for different gestures. For a local optimum k^*, we propose the following conditions to check whether it is the global optimum.

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

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

Condition 1 is a necessary condition for the global optimum. If Condition 1 is not satisfied, we continue to search and check the next local optimum. In Condition 2, if the threshold is set strict (i.e., τ is low), it fails to consider the possibility of user's poor performance even if the user has completed the gesture. If the threshold is set loose (i.e., τ is high), $T(k^*) < \tau$ may be satisfied at some local optimums before the user completes the gesture. To solve this problem, we propose a dual-threshold strategy as follows. In Condition 2, a strict threshold τ_S is used. Therefore, Condition 2 is used to detect the global optimum when the user is following the PT avatar accurately. If a local optimum satisfies both Condition 1 and Condition 2, it can be estimated as the global optimum. If only Condition 1 is satisfied and Condition 2 is not satisfied, we further use the following method to check whether k^* may be the endpoint of the user's gesture. If k^* is the global optimum n_1, B_1 is the best match with A_1. When the user completes one gesture, he/she may stay in the ending posture for several frames, so the following frames $\{b_{n_1+1}, b_{n_1+2}, \cdots\}$ will be quite close to bn1. Based on the above observation, we propose the following empirical evidence. For the global optimum n_1, all of its following r frames $\{b_{n_1+1}, b_{n_1+2}, \cdots, b_{n_1+r}\}$ tend to be aligned with a_{m_1} in DTW. In other words, for frame $n_1 + j (j = 1, 2, ..., r)$, Equation (2.6) becomes

$$S'(m_1, n_1 + j) = D(m_1, n_1 + j) + S'(m_1, n_1 + j - 1). \tag{2.8}$$

For the r frames following a local optimum k^*, we calculate the DTW distances $S_{true} = \{S(m_1, k^* + 1), S(m_1, k^* + 2), \cdots, S(m_1, k^* + r)\}$. In the meantime, we compute $S_{assumption} = \{S'(m_1, k^* + 1), S'(m_1, k^* + 2), \cdots, S'(m_1, k^* + r)\}$ using Equation (2.8) The relative error between S_{true} and $S_{assumption}$ is

$$error = |S_{assumption} - S_{true}|./S_{true}. \tag{2.9}$$

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

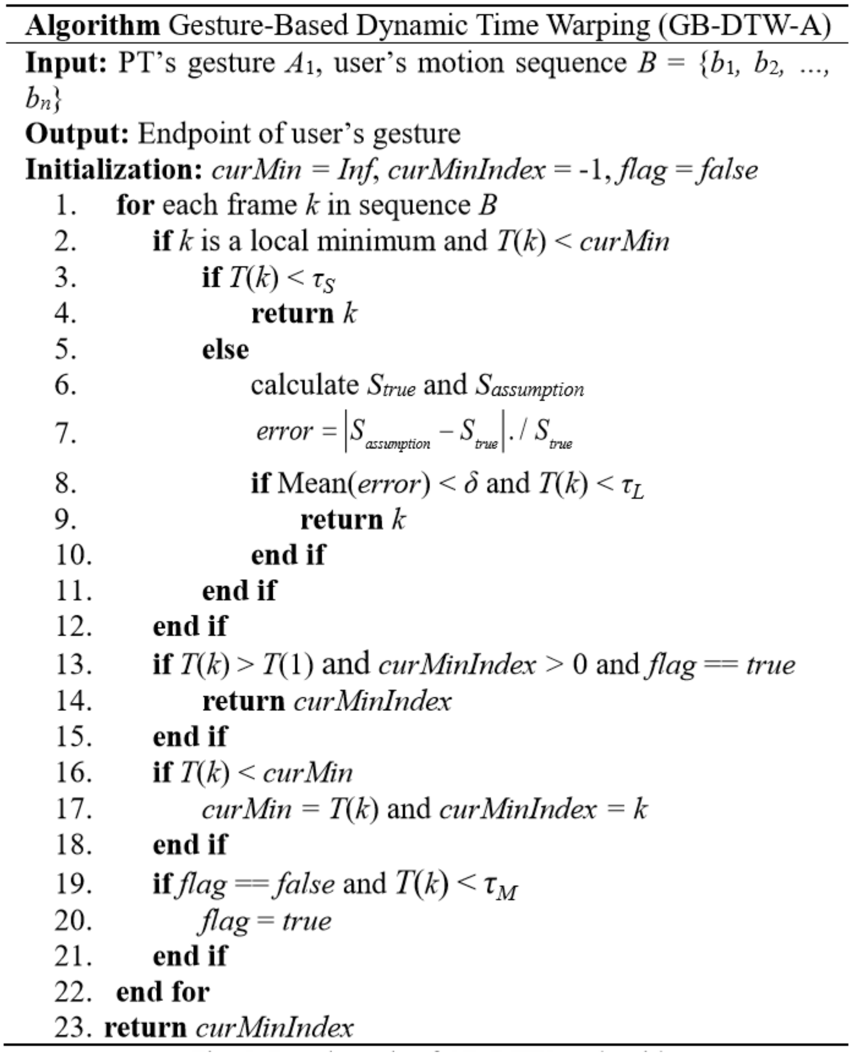
Condition 3: The relative error between S_{true} and $S_{assumption}$ is below an error tolerance threshold δ, i.e., $Mean(error) < \delta$. Besides, the normalized distance between A_1 and B_1 is below a loose threshold τ_L, i.e., $T(k^*) < \tau_L$.

Condition 3 is used to detect the global optimum for the user's poor performance. When the user performs the task, the normalized distance T(k) is calculated for each frame k. For any local optimum k^*, it is estimated as the global optimum if it satisfies Condition 1 and 2. If Condition 2 is not satisfied, Condition 3 is further used to test it. However, it is still possible that a true global optimum n1 does not meet Condition 2 or 3. If we continue searching the following frames after n^1, $T(k)$ will keep increasing and we cannot obtain the correct segmentation result even until the end of the task. To stop the searching timely, we propose Condition 4 to decide whether the current frame k is behind the global optimum n_1.

Condition 4: $T(k) > T(1)$ and there exists $k_0 < k$ such that $T(k_0) < \tau_M$. In Condition 4, $T(k_0) < \tau_M$ is used to exclude the situation where $T(k)$ may be increasing for the first several frames. If frame k satisfies Condition 4, the search should be stopped and the current global optimum (i.e., the minimum point among $T(1) \approx T(k)$) can be estimated as the global optimum. The pseudo-code for the proposed GB-DTW-A algorithm is shown in Figure 2.5.

Compared with GB-DTW0 proposed in [14], the new GB- DTW-A algorithm achieves higher segmentation accuracy and less segmentation delay. In GB-DTW0, only Condition 3 is used to test local optimums. However, the single threshold τ is sensitive to the user's performance. Figure. 2.6 shows an example where the task and motion feature are the same as Figure 2.2. Figure 2.6(a) shows the motion sequence of a PT's gesture, and Figure 2.6(b)(c) show the motion data of two users, where E_1 and E_2 are the endpoints of their gestures. User 1 follows the PT avatar accurately, so the DTW distance between the PT and User 1 is small. For the true gesture endpoint E_1, the relative error in Equation (2.9) may be high since Strue is small. In this case, the threshold τ should be higher to allow E_1 to be detected as the global optimum. User 2 is performing poorly (not following the PT avatar accurately), so the DTW distance is large. Point A is a local optimum of the DTW distance, but not the gesture endpoint. For point A, the relative error in Equation (2.9) may be small since Strue is large. To avoid mistakenly detecting A as the global optimum, τ should be set lower. Therefore, a uniform threshold τ for all users may result in segmentation errors. In contrast, the dual-threshold strategy proposed in GB-DTW-A can be used for all types of user performance, and therefore reduce the segmentation errors. Besides, the segmentation delay (i.e., the delay between the true gesture endpoint and the time when the segmentation is completed) of GB-DTW0 is at least r frames since Condition 3 needs to check r frames following the gesture endpoint. In GB-DTW-A, Condition 1 and 2 can be checked in real time without any delay. Condition 3 is checked only if Condition 2 is not satisfied. Moreover, Condition 4 provides a way to stop the searching in time when we miss the global optimum instead of searching to the end of the task (which is used by GB-DTW0). Thus GB-DTW-A also reduces the segmentation delay compared with GB-DTW0. Details about the comparison results between these two algorithms are provided in Section 2.5.2.

Using the above approach, gesture segmentation is implemented in the process of DTW. If $B_1 = \{b_1, b_2, \cdots, b_{n_1}\}$ is determined as the user's gesture related to the PT's gesture A_1, DTW can be conducted from the new starting point $(m_1 + 1, n_1 + 1)$. Figure 2.7 shows the example of applying GB-DTW-A on the same sequences as Figure 2.4. Suppose that there are four gestures in the task, segmentation allows DTW to be performed separately for each gesture. The shaded area is indicative of the computation cost for each gesture. For each gesture, Condition 1 and 2 can be checked on each local optimum in constant time. For a task with g gestures, each PT's gesture contains m/g frames and each user's gesture contains n/g frames on average. The complexity of GB-DTW- A on each gesture is $O(mn/g^2)$. For Condition 3, r more frames following the local optimum need to be tested. The extra complexity to test local optimum is $O(mr/g)$. So the average complexity of GB-DTW-A is



```
**Algorithm** Gesture-Based Dynamic Time Warping (GB-DTW-A)
**Input:** PT's gesture $A_1$, user's motion sequence $B = \{b_1, b_2, \ldots, b_n\}$
**Output:** Endpoint of user's gesture
**Initialization:** $curMin = Inf$, $curMinIndex = -1$, $flag = false$
  for each frame $k$ in sequence $B$
      if $k$ is a local minimum and $T(k) < curMin$
          if $T(k) < \tau_S$
              return $k$
          else
              calculate $S_{true}$ and $S_{assumption}$
              $error = \left|S_{assumption} - S_{true}\right| ./ S_{true}$
              if Mean($error$) $< \delta$ and $T(k) < \tau_L$
                  return $k$
              end if
          end if
      end if
      if $T(k) > T(1)$ and $curMinIndex > 0$ and $flag == true$
          return $curMinIndex$
      end if
      if $T(k) < curMin$
          $curMin = T(k)$ and $curMinIndex = k$
      end if
      if $flag == false$ and $T(k) < \tau_M$
          $flag = true$
      end if
  end for
return $curMinIndex$
```

Figure 2.5: Psuedo-code of GB-DTW-A algorithm.

$$O(g \times (\frac{mn}{g^2} + \frac{mr}{g})) = O(m(\frac{n}{g} + r)) = O(\frac{mn}{g}) \ll O(mn). \tag{2.10}$$

When the number of gestures g in the sequence is large, the proposed GB-DTW-A algorithm can significantly decrease the computation complexity compared to classical DTW on the entire sequence. If real-time detection fails, which means that the true global optimum does not meet Condition 2 or 3, Condition 4 is used to break the search and output the correct segmentation result, although with some delay. In this case, the computation complexity increases.

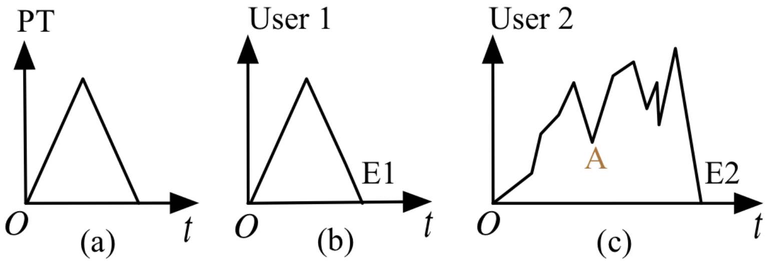


Figure 2.6: (a) PT's motion sequence. (b) User 1's motion sequence with accurate performance. (c) User 2's motion sequence with poor performance. A is a local optimum of the DTW distance.

If the segmentation is delayed to the end of the entire task in the worst case, the complexity becomes $O(mn)$. However, it is shown in Section 2.5.2 that this worst situation happens very rarely. In most cases, the segmentation delay is low and the complexity is close to $O(mn/g)$.

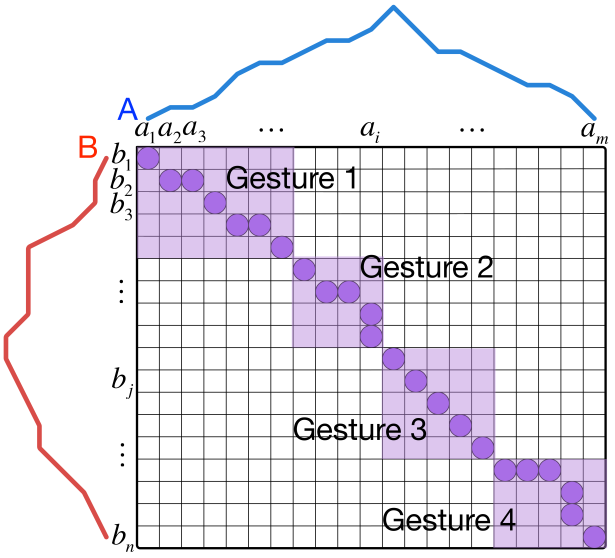


Figure 2.7: Average computation complexity of GB-DTW-A in a task of four gestures.

2.4.3 GB-DTW-A Based User Performance Evaluation

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

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

For each criterion in a task (see examples in Table 2.1), we denote $A = a_1, a_2, \cdots, a_m$ as the PT's motion data and $B = \{b_1, b_2, \cdots, b_n\}$ as the user's motion data. An optimal path $P = \{p_1, p_2, \cdots, p_q\}$ which indicates the optimal alignment between A and B has been calculated by applying GB-DTW on the time- varying motion data.

To measure the user's error, first we need to discuss different alignment types in P. We define the monotonicity of a subsequence $A^* = \{a_i, a_{i+1}, \cdots, a_{i+w-1}\}$ as follows. If all the elements in A^* are monotonic (i.e. keep increasing or decreasing) then $A*$ is monotonic, otherwise it is non- monotonic. When multiple PT frames $A^* = \{a_i, a_{i+1}, \cdots, a_{i+w-1}\}$ are aligned with one single user frame b_j, there are two different cases. (a) If $A^*$is monotonic, it means that the effects of multiple frames in A^* are similar to the effect of b_j, which indicates that the user moved faster than the PT avatar at that time. (b) If A^* is non-monotonic, it means that some back and forth PT movements are simplified as one single frame b_j in the user's performance, thus the user's movement is incomplete for this back and forth motion. Similarly, if one single PT frame is aligned with multiple user frames, we can judge whether the user is slower or overdoes the movement. (Note that the cause for the user to be slow might also be due to receiving the training video delayed due to the wireless network, that is, effect of network delay.) Table 2.2 and Figure 2.8 illustrates the five alignment types in DTW. For example, in type 1 the user performs faster than the PT avatar so monotonic PT subsequence $\{a_3, a_4\}$ is aligned with one single user frame b4. In type 4 the user's movement does not reach the required amplitude, so non-monotonic PT subsequence $\{a_{17}, a_{18}, a_{19}\}$is aligned with one single user frame b_{21}. Type 5 represents the basic case where one PT frame is aligned with one user frame.

Table 2.2: Five alignment types in DTW.

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

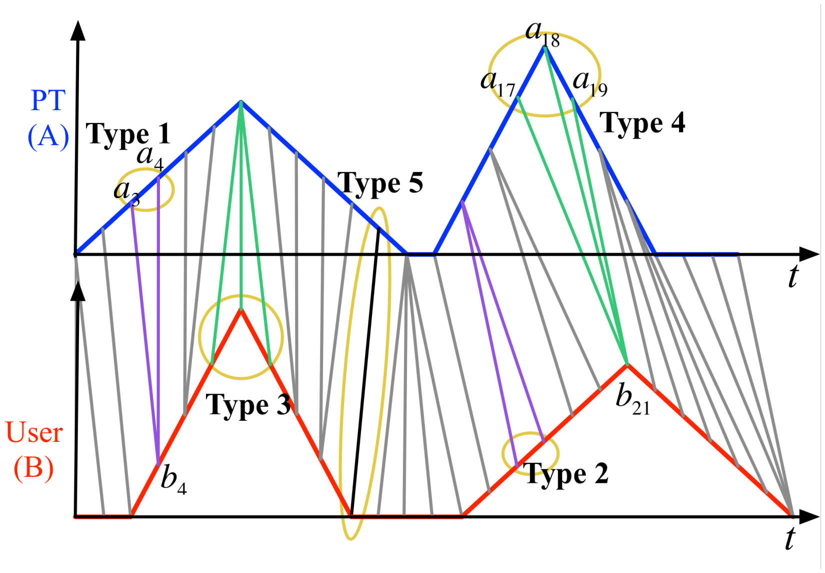


Figure 2.8: Five alignment types in DTW: 1) The user moves faster. 2) The user moves slowly. 3) User's overdone motion. 4) User's incomplete motion. 5) Basic case where one PT frame is aligned with one user frame.

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

$$e_{frame} = \|a_i - b_j\|. \tag{2.11}$$

However, if PT frame a_i is aligned with multiple user frames $B^* = \{b_j, b_{j+1}, \cdots, b_{j+w-1}\}$, the difference between the two sequences will be counted several times according to Equation (2.4.3). In this case we should revise Equation (2.4.3) to count in the user error for only once based on the alignment types in Table 2.2 and Figure 2.8. If B^* is monotonic (i.e., type 2), the user performs slower than the PT avatar. For most physical therapy tasks, user's speed is not important. (Tasks for which speed is important are not discussed in this chapter.) Only the average user error should be counted, and equation can be revised as

$$\hat{e}_{frame} = \|a_i - (\frac{1}{w}\sum_{r=0}^{w-1} b_{j+r})\|. \tag{2.12}$$

If B^* is non-monotonic (i.e., type 3) which represents the user's overdone movements, the largest user error needs to be counted, and Equation (2.4.3) can be revised as

$$\widetilde{e}_{frame} = \max_{0\leq r\leq w-1} \|a_i - b_{j+r}\|. \tag{2.13}$$

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

(2) Overall Score Estimation

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

To obtain the score estimation model, a subjective study is needed where the proposed system calculates the error vector e and our PT collaborator gives a true score s for each subject. Given the error vectors $\{e_1, e_2, \cdots, e_i, \cdots, e_N\}$ and the corresponding scores

$\{s_1, s_2, \cdots, s_i, \cdots, s_N\} (s_i \in [0, 10])$ for N samples, our goal is to find an optimal function $h(e)$ so that $s_i \approx h(e_i)$. Here we choose h to be linear and include constant 1 in e_i as the bias term. Thus h can be represented as

$$h(e) = \beta^T e. \tag{2.14}$$

We use linear regression [33] to estimate the optimal β^* as

$$\beta^* = (X^T X)^{-1} X^T y, \tag{2.15}$$

where

$$X = (e_1, e_2, \cdots, e_N)^T, y = (s_1, s_2, \cdots, s_N)^T. \tag{2.16}$$

From Equation (2.15) the optimal parameter β^* can be calculated from all the scores given by our PT collaborator and error vectors in the training set. Then this optimal function $h(e)$ can be used to estimate the overall score for any new user performance.

2.4.4 Real-Time Guidance and Satisfactory Score

In order to help the user improve performance accuracy, we propose a replay system which highlights the user's error and provides visual and textual guidance for the user. Figure 2.9 shows a screenshot of the guidance system for the leg lift task. The overall score for the user's performance is shown on the upper left corner of the screen. Two avatars replay two views of the user's movements, with the view angles determined by the task to better show the user's error. In Figure 2.9, the left avatar shows the side view and the right avatar shows the mirrored view. For each gesture, the user's motion data on each criterion are compared with the corresponding PT's motion data. If the user's error on a criterion is above the error threshold defined by the PT (see Table 2.1), the guidance video will be slowed down, and visual/textual guidance is provided for the user to calibrate his/her movements.

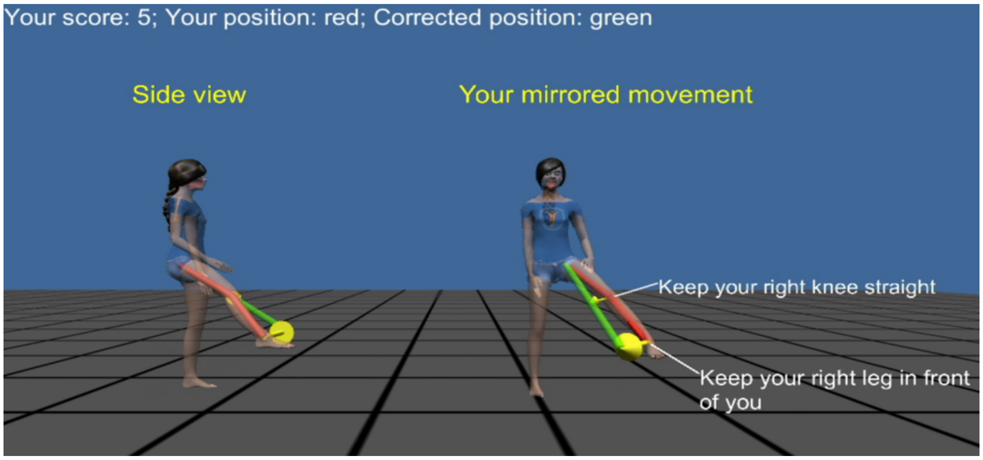

Your score: 5; Your position: red; Corrected position: green
Side view
Your mirrored movement
Keep your right knee straight
Keep your right leg in front of you

Figure 2.9: Examples of textual and visual guidance in the leg lift task. Left avatar: side view. Right Avatar: mirrored view. User's incorrect body parts: red. Corrected position: green. Textual information is placed beside the body.

Visual guidance uses colored cylinders to label the user's incorrect body positions and the correct positions. Incorrect body positions are rendered in red and the corrected positions are rendered in green so user can see the clear difference. In addition, directional arrows rendered in yellow will give further guidance on how to correct this movement. Textual guidance is provided beside the corresponding body parts to instruct the user. There are two types of textual guidance: qualitative and quantitative textual guidance. Qualitative textual guidance gives only general instructions on how to calibrate incorrect motion (e.g., "bring your right leg higher"), while quantitative textual guidance provides detailed instructions on the quantitative error (e.g., "bring your right leg higher by 20 degrees"). Quantitative guidance is important for the user to make right calibrations and avoid over corrections. However, when textual guidance is provided together with visual guidance, qualitative textual guidance may be sufficient since visual guidance already gives the user intuitive instructions about the quantitative error. To determine which kind of guidance is most helpful for the user, we have conducted subjective tests, whose results are shown in Section 2.5.4.

In addition, there are multiple choices for the timing of providing guidance. For example,

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

Hence, we believe that post-gesture guidance is the most helpful in the proposed system. Real-time gesture segmentation has been achieved by the proposed GB-DTW-A algorithm. To determine whether to provide guidance or continue training, a satisfactory score is set by our PT collaborator (which will be discussed in Section 2.5). Scores above the satisfactory score means that the user passes the gesture and can progress to the next gesture. Otherwise, the system will pause the training and provide guidance for this gesture.

2.5 Experimental Results

We conducted experiments based on the testbed (shown in Figure 2.10) we have developed to emulate the system architecture in Figure 2.1. The cloud server is running on a desktop with a quad core 3.1GHz CPU and 8GB RAM, and the user device is a laptop PC with a dual core 2.5GHz CPU and 4GB RAM. The network connection between the cloud server and the mobile laptop is emulated using a network emulator (Linktropy [34]), which can be programmed to emulate different wireless network profiles. All the experiments were conducted with the assistance of a licensed PT who specializes in movement disorder population with a background in orthopedics and fitness.

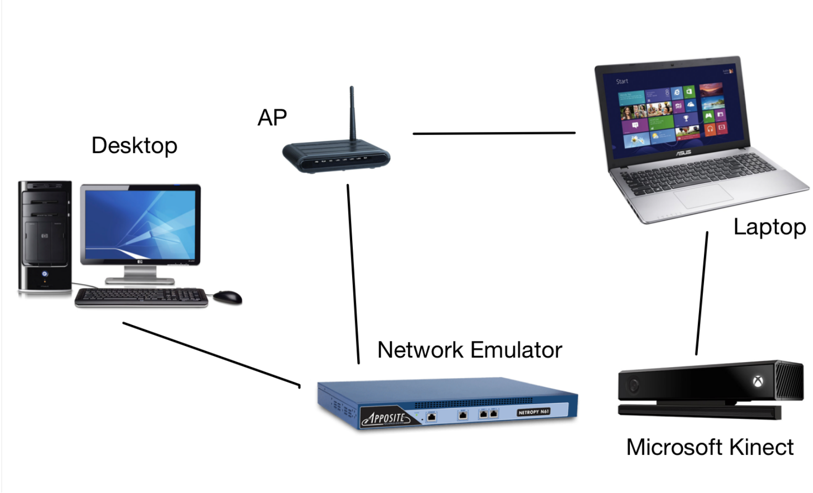


Figure 2.10: Experiment testbed.

2.5.1 Experiments to Validate Data Alignment Approach

To validate the proposed data alignment and gesture segmentation approach, the tested task is shoulder abduction and adduction (shown in Figure 2.2(a), criteria and motion features are shown in 2.3) with different target heights for five times. The PT's motion data for the five gestures are shown as the blue curve in Figure 2.11. Only the left shoulder angle is shown here for simplicity.

Four users (User A, B, C and D) with different motion abilities were invited as subjects in the experiment. They tried to follow the PT avatar's movements by watching the training video which was transmitted through the network emulator to the laptop. Each user was tested under ideal network condition (without any bandwidth constraint) and non-ideal network condition

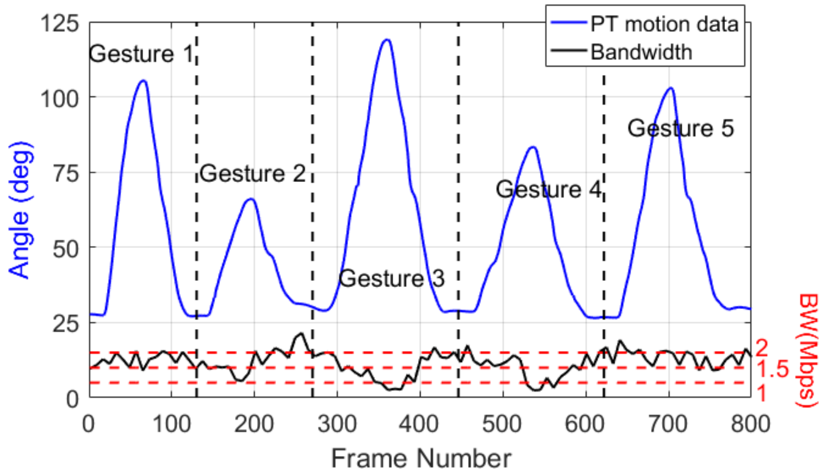


Figure 2.11: PT's motion data (i.e., left shoulder angle) and the bandwidth profile.

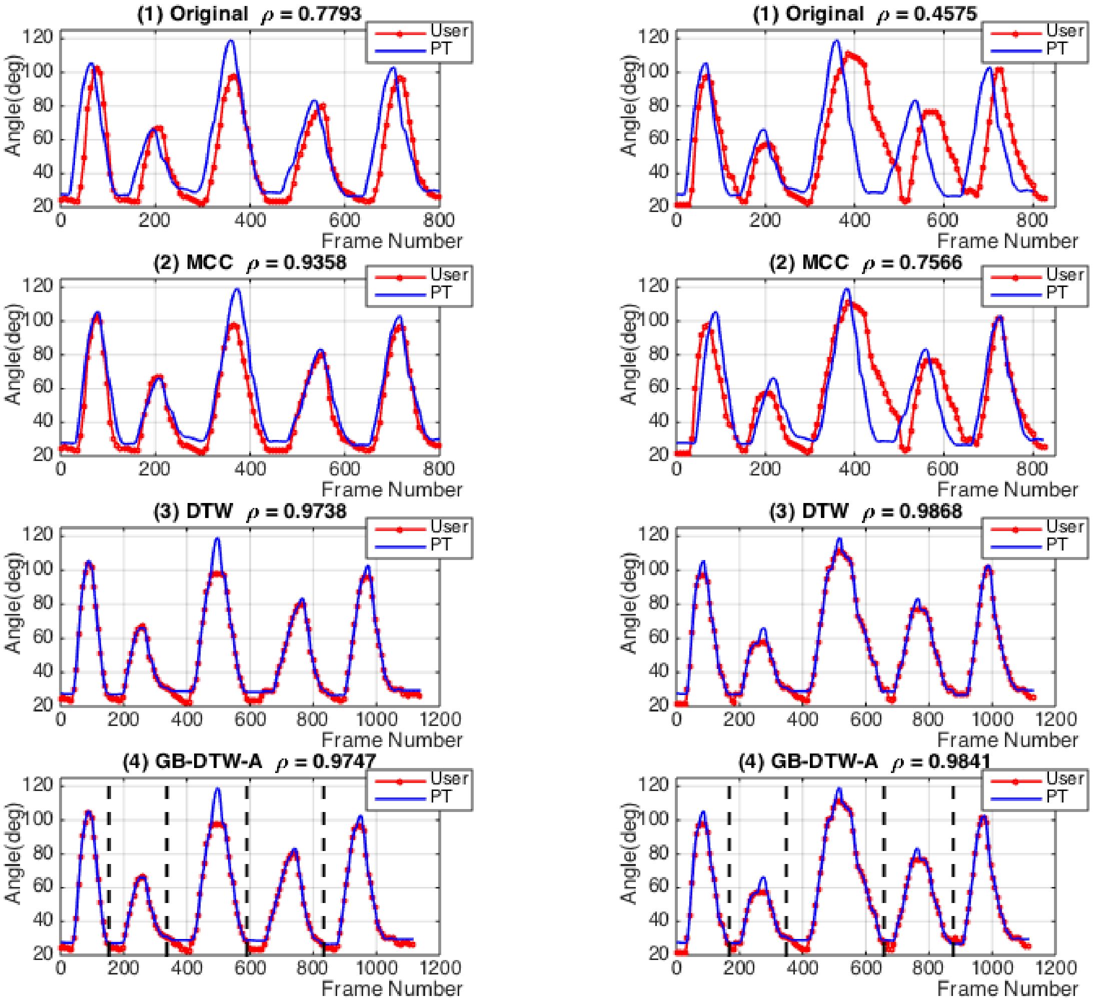
(limited by a bandwidth profile to simulate the downlink network). The bandwidth profile is shown as the black solid curve in Figure 2.11 and it was repeated for each user using the network emulator. It can be observed that the bandwidth is relatively lower at the third and fourth gestures. Then we use three different techniques: 1) tradition method of MCC, 2) classical DTW on the entire motion sequences, and 3) GB-DTW-A, to align the motion sequences of the PT and the user. For the GB-DTW-A algorithm, the double thresholds $\{\tau_S, \tau_L\}$ are set as $\{0.1, 0.5\}$ and $\tau_M = 0.5$, $r = 20$, $\delta = 5\%$.The alignment results of user A are shown in Figure 2.12. In each figure, we plot the motion data of the PT and the user, with the x-axis showing the frame number and the y-axis showing the tested shoulder angle. The vertical dashed lines in GB-DTW-A show the gesture segmentation results. In the two DTW algorithms, when multiple frames in one sequence are aligned with one single frame in the other sequence, the single frame is repeated for several times to show the alignment results. From Figure 2.12 we can see that the user performs worse with fluctuating bandwidth than ideal network condition due to the network delay. Especially at the third and fourth gestures when bandwidth is limited, he/she cannot follow the PT avatar and performs more slowly. To quantify the alignment results, we calculate the correlation coefficient /rho of the aligned sequences x and y in each method as

Table 2.3: Motion features and criteria of shoulder abduction and adduction, leg lift and jumping jack.

Task	Feature Type	Criteria	Feature
Shoulder abduction and adduction	Time-varying ($N_v = 1$)	"Raise the arm to the required height"	Angle between the arm and the vertical direction: set by the human PT (e.g., 90°)
	Constraint ($N_c = 1$)	"Keep the elbow straight"	Angle between the upper arm and lower arm: 180°
Leg lift (right)	Time-varying ($N_v = 1$)	"Lift right leg to the required height"	Angle between the right leg and the vertical direction: set by the human PT (e.g.,60°)
	Constraint ($N_c = 4$)	"Keep the trunk upright"	Angle between the trunk and the vertical direction: 0°
		"Keep pelvis level"	Angle between the pelvis and the horizontal direction: 0°
		"Knee right knee straight"	Angle between the right thigh and shank: 180°
		"Keep right leg in front of the body"	Angle between the right leg and the patient's right direction: 90°
Jumping jack	Time-varying ($N_v = 2$)	"Raise left arm beyond the head"	Angle between left arm and the vertical direction: 120°
		"Raise right arm beyond the head"	Angle between right arm and the vertical direction: 120°
	Constraint ($N_c = 5$)	"Keep left and right arm symmetrical"	Difference between the two time-varying features: 0°
		"Keep left arm aligned with the trunk"	Angle between the left arm and the body plane: 0°
		"Keep right arm aligned with the trunk"	Angle between the right arm and the body plane: 0°
		"Keep left elbow straight"	Angle between the left upper arm and lower arm: 180°
		Angle between the left upper arm and lower arm: 1800°	Angle between the right upper arm and lower arm: 180°

$$\rho = \frac{E[(x-\bar{x})](y-\bar{y})]}{\sqrt{\sigma_x^2 \sigma_y^2}}, \tag{2.17}$$

where $\bar{x}, \bar{y}$ are the means of x, y and σ_x^2, σ_y^2 are the variances. High correlation coefficient indicates



User A (Ideal Network)

User A (Non-Ideal Network)

Figure 2.12: Data alignment results for User A under ideal and non-ideal network conditions. (1) Original misaligned motion sequences. (2) MCC. (3) classical DTW. (4) GB-DTW-A and gesture segmentation results.

that the two sequences are aligned better. The correlation coefficients for each user using different methods are shown in Table 2.4. Comparing the three methods, it can be concluded that when the user follows the PT avatar quite well and there is only time shift delay, the traditional method of MCC gives high correlation coefficients ($\rho > 0.85$). However, when the network condition is not ideal and therefore the training video is delayed, or when the user cannot follow the PT avatar due to his/her motion ability, the user's motion data are distorted. In this case the two DTW algorithms perform much better (ρ > 0.95) than MCC (ρ < 0.80). For DTW and GB-DTW-A, their alignment results are quite close and both of their correlation coefficients are more than 0.95. Figure 2.13 shows the running time of DTW and GB-DTW-A on the four users under ideal and non-ideal network conditions. We can see that GB-DTW-A needs significantly less time compared with DTW to align the two sequences, which validates our deduction in Equation (2.10). Therefore, the proposed GB- DTW-A outperforms other alignment methods as well as enable real-time guidance with reduced computation complexity.

Table 2.4: Correlation coefficients for user A, B, C, and D using dfferent alignment methods under ideal and non-ideal network conditions.

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

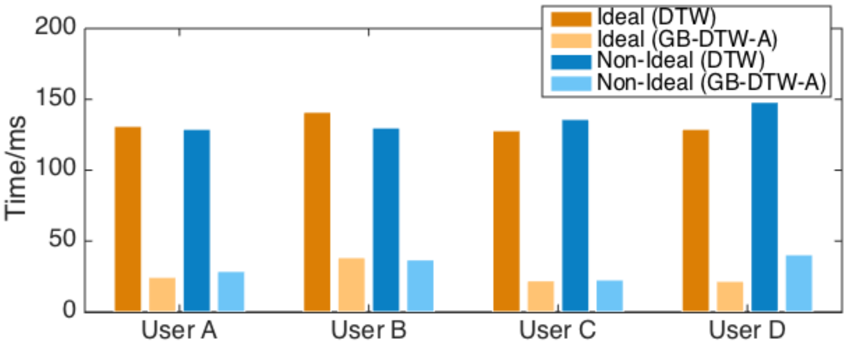


Figure 2.13: Running time of DTW and GB-DTW-A under ideal and non-ideal network conditions.

2.5.2 Experiments to Compare GB-DTW0 and GB-DTW-A

Wireless networks can be associated with significant jitter (variations in network delay). Jitter can exacerbate the motion data misalignment problem due to network delay, and challenge the performance of the data alignment and segmentation algorithms GB-DTW0 [14] and GB-DTW-A. Hence, we conduct experiments to compare the performance of the two algorithms in the presence of jitter. We emulate the condition where the user follows the PT avatar accurately, but his/her motion sequence is affected by jitter. By using this "perfect" user, the motion data misalignment is completely caused by jitter. Therefore, the effectiveness of the two algorithms can be tested by checking whether they can achieve "perfect" alignment result (correlation coefficient close to 1). In our experiments, the motion sequence of the "perfect" user is created by delaying each frame of the original PT's motion sequence shown in Figure 2.11 by Δt. (Frames are not reordered even if subjected to differing delays.) Δt follows a positive truncated normal distribution (i.e., $\Delta t \sim |N(0,\sigma^2)|$), and the mean of Δt is

$$\mu_{\Delta t} = \int_0^\infty \Delta t \frac{2}{\sqrt{2\pi}\sigma} e^{-\frac{\Delta t^2}{2\sigma^2}} d\Delta t = \sqrt{2/\pi}\sigma. \tag{2.18}$$

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

Correlation Coefficient (CC): see Equation (2.17).

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

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

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

Results are shown in Figure 2.14, with the x-axis showing $\mu_{\Delta t}$ and y-axis showing CC, UE, SE, and SD in the four sub-figures respectively. Since each user motion sequence contains only network delay, the user's performance can be considered "perfect" and thus CC should be close to 1 and UE should be close to 0. Smaller SE indicates more accurate segmentation and smaller SD means more real-time segmentation. From Figure 2.14 it can be concluded that, when the jitter is low (i.e., $\mu_{\Delta t} < 2s$), both GB-DTW0 and GB-DTW-A achieve good segmentation and alignment results. Note that the SD result of GB-DTW0 is always larger than 20 frames because Condition 3 is always used to check r frames following the gesture endpoint. When jitter is higher (i.e., $\mu_{\Delta t} > 4s$), GB-DTW-A shows superiority over GB-DTW0, especially in maintaining low SE and SD. The average number of CC, UE, SE, and SD for GB-DTW0 and GB-DTW-A are shown in Table 2.5. We can observe significant improvements achieved by the new algorithm GB-DTW-A compared to GB-DTW0 [14]], especially in reducing estimation of user error (lower UE), enhancing segmentation accuracy (lower SE), and making segmentation real-time (lower SD). Note that the segmentation delay (SD) achieved by the new algorithm GB-DTW-A is only 11 frames on average, compared to an average of 39 frames for GB-DTW0, and never higher than 40 frames. The low SD numbers achieved by GB-DTW-A validates that the computation complexity of GB-DTW-A is close to $O(mn/g)$ in most cases, and since it never has to search till the end of the user sequence, it shows that it never reaches the worst-case computation complexity of $O(mn)$ (section 2.4.2).

Table 2.5: Average improvements of GB-DTW-A compared to GB-DTW0.

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

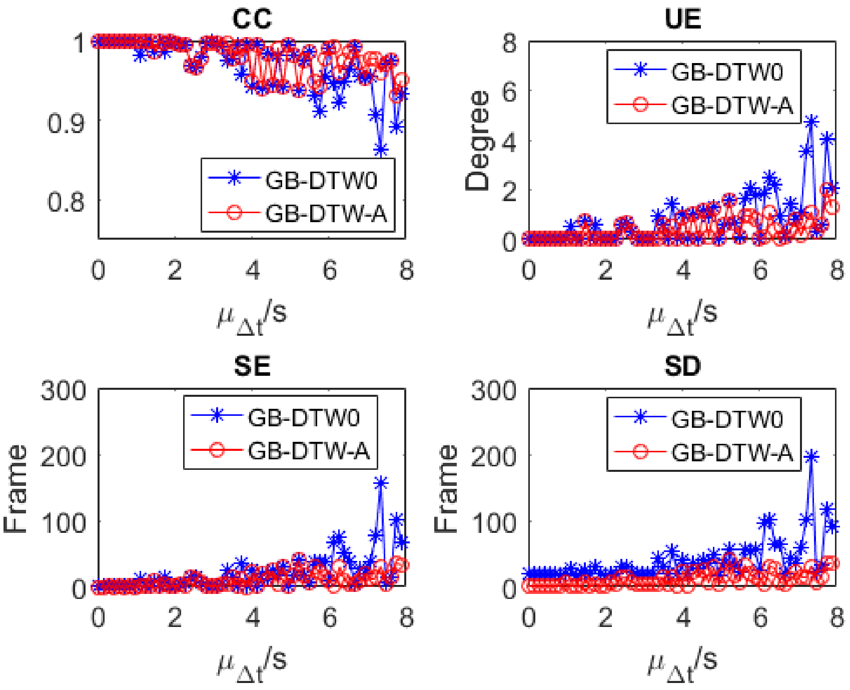


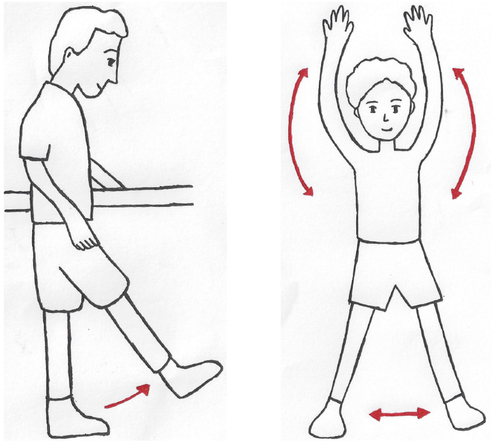
Figure 2.14: Comparison between GB-DTW0 and GB-DTW-A. The four sub- figures show results of correlation coefficient (CC), user error (UE), segmentation error (SE), and segmentation delay (SD).

2.5.3 Experiments to Estimate Overall User Score

As discussed in Section 2.4.3, the optimal function $h(e)$ for each task can be estimated by applying linear regression on training samples. In this experiment, the tested tasks are leg lift and jumping jack which are shown in Figure 2.15. Motion features and criteria for each task are shown in Table 2.3.

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

All the samples are randomly divided into a training set (including 42 samples) and a test



(a) (b)

Figure 2.15: (a) Leg lift. (b) Jumping jack.

set (including 18 samples). For the training set, Equation (2.15) is used to train the samples and calculate $h(e)$. Then we apply the optimal function $h(e)$ on the test set. The results are shown in Figure 2.16, with the x-axis showing the real score s_{PT} given by the PT and the y-axis showing the estimated score sestimated using $h(e)$. The mean absolute error (MAE) between s_{PT} and $s_{estimated}$ is calculated and shown in Figure 2.16. Samples on the diagonal line $s_{PT} = s_{estimated}$ means that the estimated score is the same as the real score without any error. The two dotted lines $s_{PT} = s_{estimated} \pm 1$ define the diagonal area for which the estimation error is below 1. (We choose 1 as the error threshold since most scores given by the PT are integers, for which 1 is the minimum error.) We can see that most of the test samples lie in the diagonal area, which means that the evaluation models are accurate. Besides, using $h(e)$ to evaluate the patients is superior because the intra-rater reliability [35] of a human performing movement analysis without any analytical tools besides eye site shows increased variability. By utilizing the system to analyze the movements there is a more uniform scoring and increased intra-rater reliability.

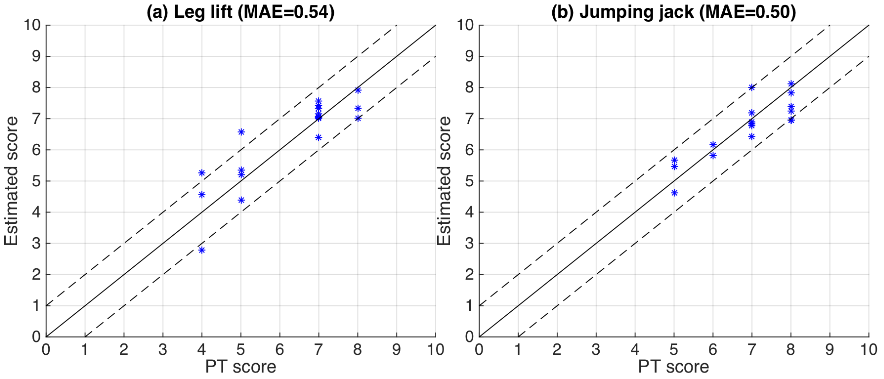


Figure 2.16: Estimated score vs. human PT's real score and the mean absolute error (MAE) for (a) leg lift and (b) jumping jack.

2.5.4 Effectiveness of Visual and Textual Guidance

As discussed in Section 2.4.4, visual and textual guidance can be provided after each gesture according to the user's performance. The satisfactory score is set as 7 by our PT collaborator in order to allow for some intrinsic error correction, which would allow for increased learning of the task. If the threshold is set too low, the patients would obtain a passing score too easily and not have the correct amount of feedback to properly correct the deficits in his/her movements. If the score is set too high, it might discourage the patients from trying their best and create a negative mindset, resulting in a reduction in retention.

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

In the test, 28 subjects (aged 17 $\sim$ 38, 14 males, 14 females) were invited to perform two training tasks (leg lift and jumping jack) using the proposed system. To ensure the same initial average score of each group, groups were assigned after the first attempt of each subject. Each subject performed each task four times and the average score of each group is calculated. Figure 2.17 shows the average performance and 90% confidence intervals (black vertical lines) of each group, with each group represented by a different color. The red dotted curve shows the satisfactory score.

From Figure 2.17 we can see that the average scores on the first attempt in each group are similar, which ensures similar initial ability of each group. We also make the following important conclusions from the results. While scores for people in group N (without any kind of guidance) fluctuates with large confidence intervals, and may or may not reach the satisfactory score, using each type of real-time guidance helps the users improve performance, though with varying effectiveness. People in group V (who get visual guidance) and group T (who get quantitative textual guidance) reach the satisfactory score 7 after the fourth attempt. On the other hand, the results show that the combination of visual and textual guidance is the most helpful: it helps users in group VT reach score 7 after only the second attempt.

2.5.5 Performance Validation Using Real Cloud Environment

To validate the performance of the proposed system on a real cloud environment, we implemented the system on Amazon Web Services (AWS) [36]. The experiment setup is the

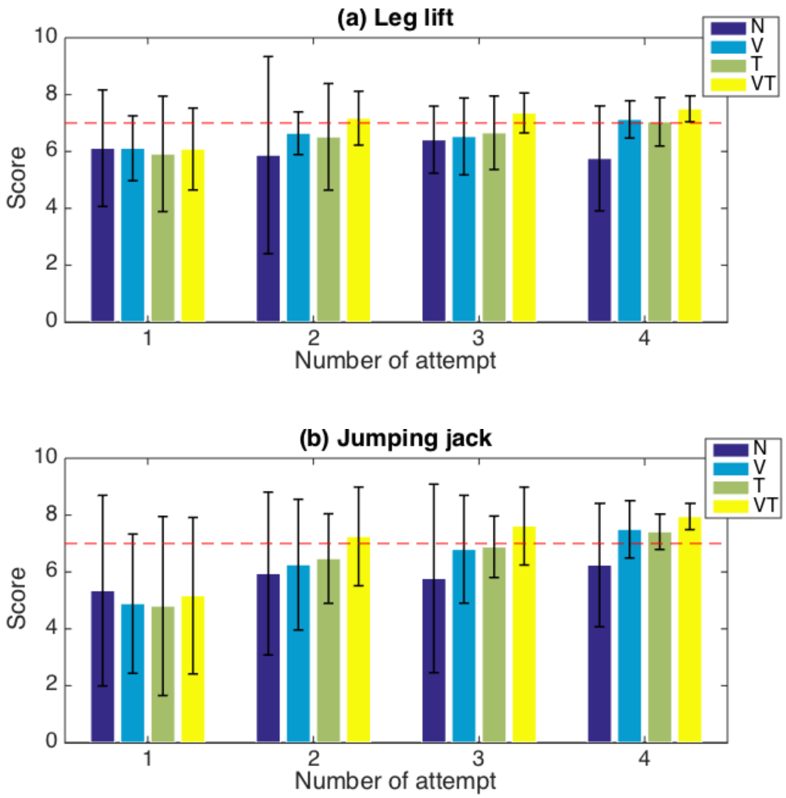


Figure 2.17: Average score of each group with vertical lines showing 90% confidence interval. (a) Leg lift. (b) Jumping jack.

same as Figure 2.10 except that the desktop and network emulator are replaced by AWS (and the real network from AWS to the user device). Specifically, we use AWS g2.2xlarge instance which provides access to one NVIDIA GRID GPU with 1,536 CUDA cores and 4GB of video memory. The CPU it provides is Intel XeonE5–2670 @2.60GHz with 15GB memory. The operating system we deploy is Windows_Server-2008-R2_SP1.

One of the concerns of having the system run on a real cloud environment is the possible impact of additional delay from the cloud to the user device. We tested the delay of the training and guidance videos under three different network conditions: 1) unloaded network (e.g., accessing our cloud-based system using home Wifi at midnight), 2) loaded network (e.g., accessing our cloud-based system using LTE network at 5pm), 3) loaded and noisy network (e.g., accessing our cloud-based system using public Wifi at 5pm). The histograms of the measured delay under each condition are shown in Figure 2.18, with the *x*-axis showing the delay and the *y*-axis showing the frequency of each value (i.e., number of occurrences of the value).

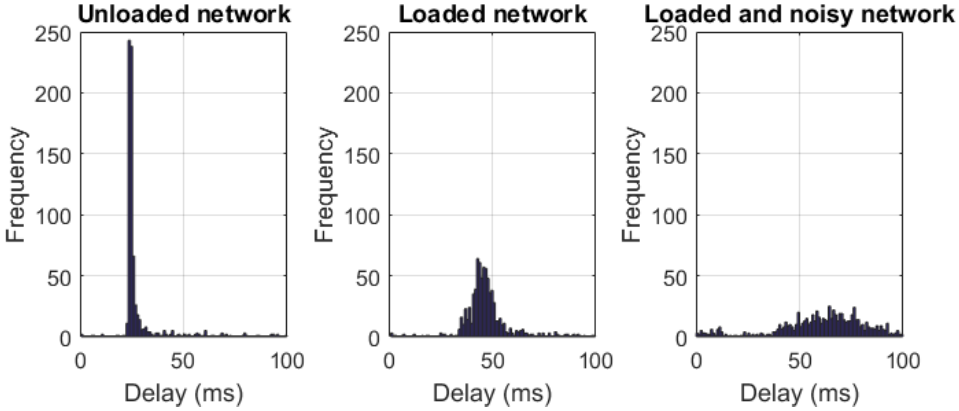


Figure 2.18: Histogram of the measured delay of avatar video from cloud (AWS) to user device under unloaded, loaded, and loaded and noisy network conditions.

The mean and Standard Deviation (STD) of the measured delay are shown in Table 2.6. When the network is unloaded, the delay is under 30ms most of the time. When the network is loaded and noisy, the delay is increased significantly but still under 100ms, which means that the video streaming from the cloud to the user's mobile device can be considered real-time in the system. Furthermore, we invited three new users to perform the shoulder abduction and adduction task using the proposed training system. The motion data alignment algorithm (i.e., GB-DTW, see Section 2.4.2) and the user performance evaluation algorithm (see Section 2.4.3) are implemented on AWS. Figure 2.19 shows the motion data alignment results. We can see that the proposed GB-DTW algorithm still works well in aligning motion data and segmenting gestures in the real cloud environment. Table 2.7 shows the running time of the alignment and evaluation algorithms on AWS. The running time of the two algorithms are under 20ms, again demonstrating their real-time nature. From all the above results, it can be concluded that the proposed system is able to provide real-time training and guidance for the user in a real cloud environment.

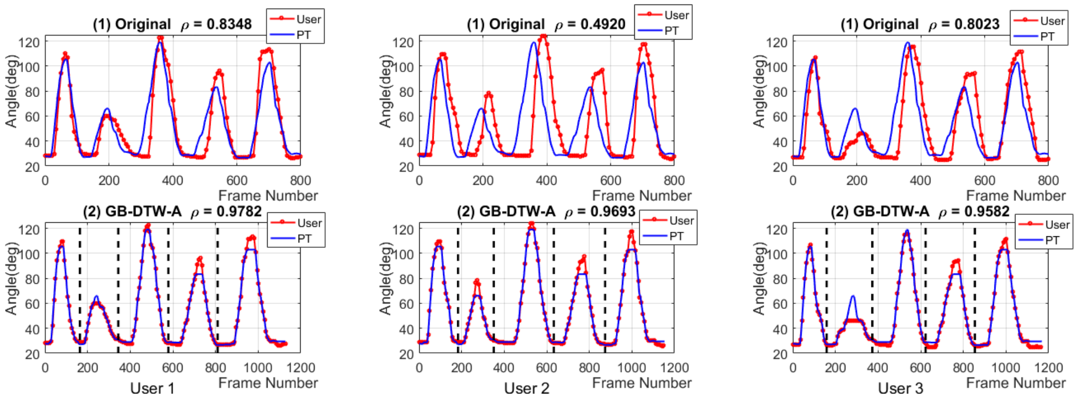


Figure 2.19: Data alignment results for User 1, 2, 3 using AWS. (1) Original misaligned motion sequences of the PT and the user. (2) Aligned sequences using GB- DTW-A and gesture segmentation.

Table 2.6: Mean and STD of delay from cloud to user device under unloaded, loaded, and noisy network conditions.

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

Table 2.7: Running time of GB-DTW-A and user performance evaluation algorithms in cloud (AWS).

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

2.6 Conclusion

In this chapter, we propose a cloud-based physical therapy monitoring and guidance system that captures and evaluates the user's performance automatically. It can also be applied to many other types of training applications, such as wellness and fitness training, and ergonomics training. To address the motion data misalignment problem as well as enable real-time evaluation, we propose the GB-DTW-A algorithm to align the motion data and segment the user's motion sequence into gestures in real time with reduced computation complexity. Experiments with multiple subjects using real network profiles show that the proposed method works better than other alignment techniques. Moreover, we provide results to demonstrate the accuracy and real-time performance of the proposed GB-DTW-A algorithm. Furthermore, the evaluation model for the user's performance is trained based on subjective test and linear regression method. Testing results show that the evaluation model is able to provide an accurate score which is quite close to the real score given by the human PT for the user's performance. Besides, the proposed guidance system can provide detailed visual and textual guidance, whose effectiveness has been validated in subjective test. Experiments using real cloud environment AWS show that the proposed system can provide real-time training and guidance for the user.

In the following chapters, we will focus on patients with PD and collect real patient data to train the virtual PT model. Moreover, the training tasks for patients with PD will be more complicated compared with the exercises discussed in this chapter. Therefore, we will also propose an enhanced model to evaluate the patient's performance. In addition, the current training system provides uniform training tasks and criteria for every patient, which may cause injury or over corrections. Thus we would like to make the criteria of each task more adaptive and personalized for patients according to their health conditions. In Chapter 3, We will propose multiple difficulty levels for each training task and develop a task recommendation model that can provide personalized training and task recommendations, like a human PT does at the clinic.

Chapter 2, in part, is from the material as it appears in proceedings of ACM conference on Wireless Health 2015. Wenchuan Wei; Yao Lu; Catherine D. Printz; Sujit Dey. and in Multimedia Tools and Applications 2017. Wenchuan Wei; Yao Lu; Eric Rhoden; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

Chapter 3

Machine Learning-Based Patient Action Understanding, Assessment and Task Recommendation

3.1 Introduction

In Chapter 2, the proposed physical therapy monitoring and guidance system is designed for general physical therapy training. In this chapter, we will focus on patients with PD. Parkinson's disease (PD) is the most common movement disorder. It affects about 1 million people in the US and 10 million worldwide [37]. The combined direct and indirect cost of PD is estimated to be nearly USD 25 billion per year in the US alone [37]. Physical therapy is an essential treatment for patients with PD. As discussed in Chapter 2, traditional physical therapy may be expensive and inconvenient for patients with PD due to factors such as insufficient insurance coverage, impaired mobility, etc. In traditional physical therapy (shown in Figure 3.1), a PT selects the training tasks, instructs the patient on how to perform the tasks, identifies and corrects the patient's errors, and regularly updates the tasks, all in the clinic. After the PT session, the patient is expected to practice the training tasks at home by following written instructions provided by the PT. However, the patient's performance and adherence to the tasks cannot be tracked at home without the supervision of the PT. Practicing a task with incorrect technique is not only ineffective for motor learning, it may also cause injury due to the impaired mobility of patients with PD. King et al. has shown poor outcomes with unsupervised home-based exercise programs for patients with PD [38]. Furthermore, the training tasks cannot be updated until the patient's next PT visit. Continuing to practice the same training task, which may not be suitable any more for the current state of the patient, could limit the patient's progress or even reinforce motor learning in a negative way. To address these problems, several automated training systems have been developed to motivate patients and monitor their movements at home using motion capture sensors [39, 40, 41, 42, 47]. However, these systems are not aimed at performance accuracy, and cannot provide personalized task recommendation for patients.

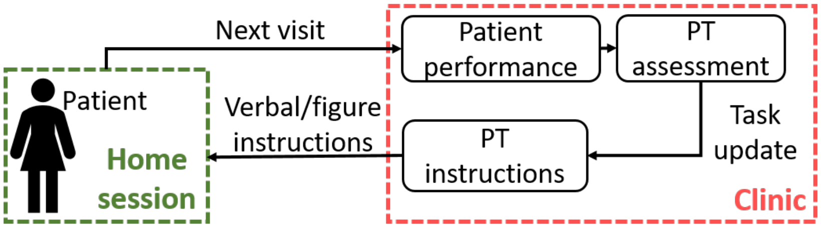


Figure 3.1: Traditional physical therapy treatment procedure.

In this chapter, we propose an on-demand virtual PT system for patients with PD. Figure 3.2 shows the proposed virtual PT system. The patient can use the cloud-based system introduced in Chapter 2, where a Kinect sensor [6] is used to capture the patient's movements and avatars are created to provide instructions and feedback. Instead of aligning the patient's motion data with PT templates to evaluate the patient performance like Chapter 2, we propose a two-phase human action understanding (TPHAU) algorithm that can understand the patient's sub-actions in performing the task and a Support Vector Model (SVM) based method to identify the patient's errors. Moreover, based on the patient's error and some subjective factors (e.g., age, discussed in

Section 3.3.4-1), a machine learning-based task recommendation model is proposed to provide automated task update recommendation for patients. Based on the recommendation results, either a new task or a guidance video will be rendered on the cloud and sent to the patient's device. The PT can remotely supervise the entire process. The proposed virtual PT system has the advantages of providing accurate, on-demand and personalized care. It has the potential of significantly reducing clinic visit requirements while offering continuous care, thereby reducing cost and expanding care for economically disadvantaged and rural patient populations. To validate the effectiveness of the proposed methods, we have collected real patient data in the Neurological Rehabilitation Clinic, UC San Diego Health. The proposed models are trained from the collected data and experimental results show that the proposed methods achieve high accuracy in patient action understanding, error identification and task recommendation.

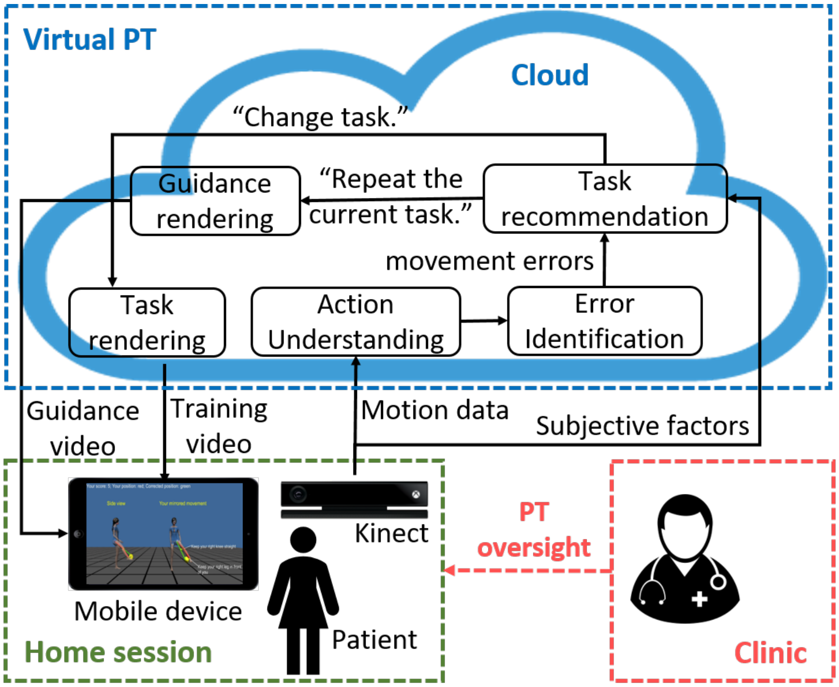


Figure 3.2: The proposed on-demand virtual PT system.

A preliminary version of this work has been reported in [45], which introduced the PT-defined training tasks and criteria for patients with PD, and proposed the action understanding and error identification methods. In [45], any task update would still need to be performed manually by the PT at the clinic. In comparison, this chapter enhances [45] to propose a machine learning-based task recommendation model to enable on-demand and personalized task recommendation for patients with PD. The task recommendations can be fully automated, or if desired, the system may require remote supervision and approval by the PT.

The remainder of this chapter is organized as follows: Section 3.2 reviews related work. Section 3.3 introduces the methods proposed in the virtual PT system. Section 3.4 presents the experimental results. Section 3.5 concludes this chapter.

3.2 Related Work

3.2.1 Automated training systems for patients with PD

While we have discussed the related work on automated training systems for physical therapy in Section , we next introduce some training systems for patient with PD. With the development of motion capture sensors, more and more sensor-based automated training systems have been developed to improve the effectiveness of home training for patients with balance and mobility problems. Hssayeni et al. [39] used wearable sensors to identify motor fluctuations in patients with PD during a variety of daily living activities. Stack et al. [40] used wearable sensors to detect subtle instability in patients with PD. However, wearable sensors attached on the body may cause extra burden to patients with PD due to their impaired mobility. Therefore, camera-based sensors were considered more convenient in monitoring the movements of patients with PD. Galna et al. [46] proved the high accuracy of the Kinect sensor in measuring clinically relevant movements in patients with PD. Galna et al. [41] and Pompeu et al. [42] designed two game-based training systems using Kinect and proved their feasibility and safety for patients with PD. However, the game-based training systems are designed to motivate the patients and cannot enable careful monitoring of desired patient performance and subsequent task recommendation like a PT does. Lin et al. [47] developed a Kinect-based rehabilitation system to assist patients with movement disorders and balance problems. However, the performance evaluation method proposed in [47] failed to consider the patient's reaction delay as it simply compared the patient's movements with the standard movement frame by frame. Most of these training systems provide uniform training for patients and cannot provide accurate evaluation, personalized feedback, and most importantly, task recommendation based on the patient's performance like a PT at the clinic. In comparison, our proposed virtual PT system provides accurate movement understanding, error identification, and personalized task recommendation, rendering our system unique. In addition, the cloud-based system can be used at any place at any time to enable on-demand virtual care, with the potential to enable personalized physical therapy with better effectiveness and compliance, while lowering cost and increasing patient participation.

3.2.2 Human action understanding

To enable automated performance evaluation, the first step is to understand the patient's movements/actions. Generally, human action understanding includes two categories: 1) Action recognition, which is the classification of an action from videos [48, 49]. However, recognizing what task the patient is performing is insufficient. We need to understand the movement details and identify the patient's movement errors. 2) Action detection/segmentation, which refers to locating actions of interest in space and/or in time [50, 51]. Most studies in this area focus on the detection of long action segments. In [50] and [51], a detected segment is considered correct if the overlap between it and the ground-truth action segment is over 40%, as this threshold is consistent with visual inspection. However, the sub-actions discussed in this chapter (see Section 3.3.1) are much shorter in time length and closer to each other (i.e., the pause between adjacent sub-actions is negligible), which makes the segmentation much more challenging. In this chapter, we propose the TPHAU algorithm to accurately detect/segment the patient's sub-actions, which will be discussed in Section 3.3.2.

3.2.3 Automated Recommendation systems

With the rapid development of artificial intelligence, more and more automated recommendation systems have been developed to enable optimized and personalized user experiences, e.g., friend recommendation in social networks [43, 63], ad recommendation [44, 64], etc. However, little research has been conducted to develop automated task recommendation systems for healthcare applications. To the best of our knowledge, we are the first to achieve automated task recommendation for patients with PD. The proposed virtual PT system is trained from real patient and PT data, thus it enables accurate and personalized task recommendation for patients with PD.

3.3 Methods

3.3.1 Kinect-based Automated Training System for Patients with Parkinson's Disease

In this section, we first introduce the training tasks selected by our PT co-author for patients with PD, then discuss how the proposed training system can identify the patient's movement errors automatically. To avoid confusion, we would like to clarify the definitions 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 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 3.3.1.

(1) Tasks and Difficulty Levels

Based on the work of King et al. [52] describing sensorimotor agility training for patients with PD, our PT co-author has selected three balance/agility based tasks: squat (SQ), forward lunge (FL) and backward lunge (BL). For each task, four difficulty levels (level $1 \sim 4$) are designed (see Table 3.1). During a traditional PT session, a patient performs a given training task at a certain difficulty level . The PT inspects the patient's performance and decides if changes to the difficulty level is needed. For example, a patient who currently performs a squat exercise may progress to a more difficult variation of the squat if the initial difficulty level becomes too easy as the patient improves. The PT's assessments are based on self-designed criteria for each task. Criteria are based on different sub-actions of a given exercise movement, which will be introduced in Section 3.3.1.

(2) Sub-actions and Criteria

For each physical therapy task, the 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 3.2, which apply to all the three tasks considered for patients with PD: SQ, FL and BL.

Table 3.1: PT-defined criteria, Kinect-captured quantities (KCQs) and applied sub-actions for Squat (SQ), Forward Lunge (FL), Backward Lunge (BL).

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

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

Levels 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

To evaluate the patient's performance in an automated and quantified way, we have defined

Table 3.2: Sub-actions in 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

some criteria for each task (i.e., the rules for evaluating the patient's performance). These criteria have been selected based on the expert PT's knowledge of compensatory movement strategies of patients with PD. For example, a common compensatory strategy that a patient with PD may use in FL is to bend the knee of the back leg, due to both strength and balance impairments. Therefore, the PT has defined "keep the back knee straight" as one of the criteria for FL. A task criterion is applicable to one or more sub-actions of the task. Table 3.3 shows the criteria defined by our PT co-author and the applied sub-actions for SQ, FL, and BL.

In the Kinect-based training system, the Kinect sensor captures 25 joints of the human skeleton with 3-D coordinates for each joint [6]. To enable automated action understanding and 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 chapter, 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 the 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 lean slightly forward thus keeping 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 the leg.

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

Figure 3.3 shows these KCQs. KCQs used in multiple tasks are shown in only one task for simplicity. The target value of each KCQ shown in Table 3.3 is either defined by the PT (e.g., $KA : 180^{\circ}$) or derived from the PT's demonstration (e.g.,$ThA : 49^{\circ}$ for small angle and 67° for large angle).

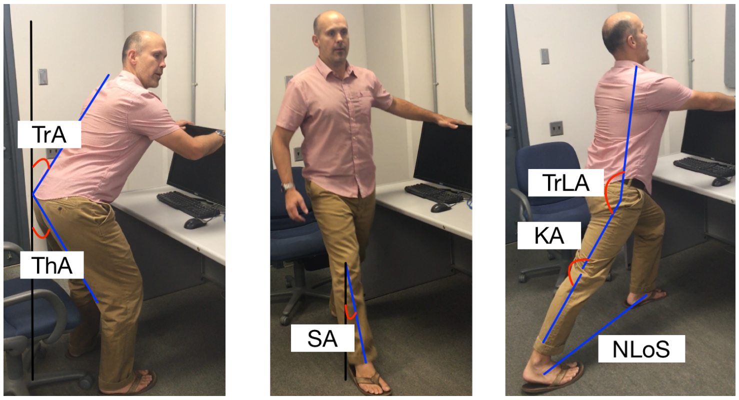


Figure 3.3: Tasks and Kinect-captured quantities (KCQs). From left to right: Squat (SQ), Forward Lunge (FL), Backward Lunge (BL).

Given the KCQs, the patient's performance can be evaluated automatically by checking the KCQs in the applied sub-actions. In Section 3.3.2, we will introduce how to segment the sub-actions in patient's movements.

3.3.2 Patient Action Understanding

Action understanding in the proposed system includes two steps: 1) Repetition detection. The patient may perform multiple repetitions on a task each time, thus we need to detect the starting point and endpoint of each repetition. 2) Sub-action segmentation, i.e., to segment the sub-actions in each repetition. To achieve this, we propose two Hidden Markov Models (HMMs) [53]: HMM-S for single repetition and HMM-M for multiple repetitions in Figure 3.4 and Figure 3.5 Details about the components of HMM are discussed in our preliminary work [45]. HMM-S consists of five hidden states S_1 to S_5. (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.) The state transfers from S_1 to S_5 and ends in S_5. For multiple repetitions, the state will transfer back to S_1 after S_4 and start a new repetition. Therefore, S_1 to S_5 are combined into one state in HMM-M. a_{ij} is the state transition probability, i.e., the probability of transferring from S_i to S_j.

Table 3.3: PT-defined criteria, Kinect-captured quantities (KCQs) and applied sub-actions for Squat (SQ), Forward Lunge (FL), Backward Lunge (BL).

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

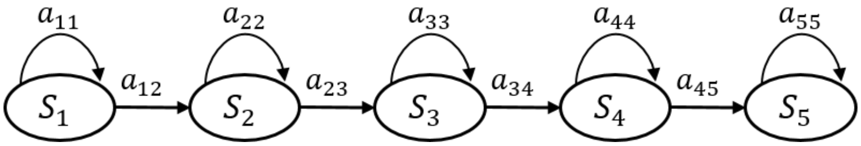


Figure 3.4: HMM-S: the HMM model for single repetition.

A key issue to be addressed for the HMM model is the HMM feature to be selected for

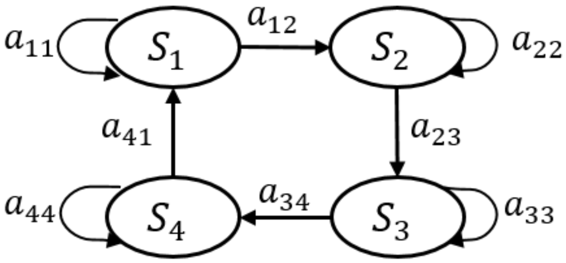


Figure 3.5: HMM-M: the HMM model for multiple repetitions.

the model. The HMM feature is the quantity we observe and use to infer the hidden states. It can be any subset of the joint coordinates, or quantities derived from the joint coordinates (like the six KCQs defined in Section 3.3.1). For the two HMM models defined in this chapter, the displacement *d* and velocity *v* of the primary moving body parts are selected as the HMM feature. In the task SQ, the patient bends his/her legs to move the hips up and down, thus *ThA* represents the movement and is used as the displacement d. In the tasks FL/BL, the patient moves one foot back and forth so *NLoS* can be used as the displacement d. The velocity *v* is calculated from d. Reasons for using the combination of *d* and *v* instead of any single variable as the HMM feature are discussed in [45].

Parameters of an HMM model include the state transition probability a_{ij}, emission probability $b_j(X)$ (i.e., the probability of observing Xunder state S_j), and the initial state distribution π_i (i.e., the probability that the Markov chain starts from state S_i). For HMM-S and HMM-M, parameters are estimated using supervised learning. Training data are collected from real patients with PD. For each training sample, five sub-actions in the movements (see Table 3.2) are manually segmented. (Note that for HMM-M, S_1 includes the manually-labelled S_1 and S_5.) The transition probability a_{ij} is calculated as

$$a_{ij} = \frac{\text{number of transitions from } S_i \text{ to } S_j}{\text{number of transitions from } S_i}, \tag{3.1}$$

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

$$b_j(X) = \sum_{c=1}^{C} w_{jc} N(\mu_{jc}, \Sigma_{jc}) \qquad (3.2)$$

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 [56]. The GMM model of each sub-action/state is trained separately using the motion data in that state.

Given the model parameters $\lambda = a_{ij}, b_j(X), \pi_i$, our goal is to infer the hidden state sequence Q from any new observation sequence O. The Viterbi algorithm [57] is a dynamic programming algorithm for finding the most likely hidden state sequence Q^* of the observation O using

$$Q^* = \arg\max_{Q} P(Q|O, \lambda) = \arg\max_{Q} P(Q, O|\lambda). \qquad (3.3)$$

Based on the Viterbi algorithm, we propose a two-phase human action understanding (TPHAU) algorithm to detect the patient's repetitions and segment sub-actions in each repetition. In the first phase, the HMM-M model is used to detect the starting point and endpoint of each repetition. Considering the difference of displacement amplitude in different patients and in different repetitions, we apply repetition-based normalization on the displacement data d of the training samples (i.e., d in each repetition are normalized into [0, 1]). For the test samples, global normalization (i.e., d of the entire performance including multiple repetitions are normalized into [0, 1]) is used since the time interval for each repetition is unknown in the first phase. Then based on the trained HMM-M model, the hidden states of the test samples can be estimated by applying the Viterbi algorithm [57] and the patient's repetitions can be further inferred. 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. Figure 3.6 shows an example. Four repetitions $R_1 \sim R_4$ are detected from the hidden state sequence.

However, noise in the motion data may cause the detection of extra repetitions. There are

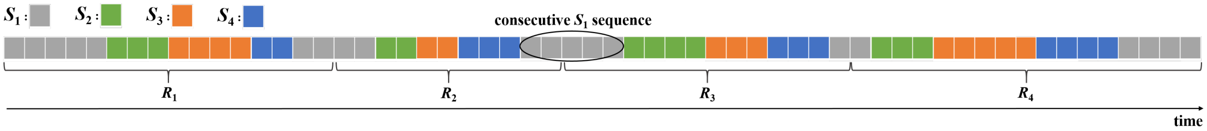


Figure 3.6: Hidden state sequence obtained from the Viterbi algorithm [57]. Four repetitions R_1, R_2, R_3, R_4 are inferred.

two types of extra repetitions: 1) noise being detected as complete repetitions, 2) recognizing one repetition as two or more. Detailed discussion about extra repetitions can be found in our preliminary work [45]. To remove the extra repetitions, we analyze the Time Length (TL), the Amplitude of Displacement (AoD) (i.e., maximum of d), and the Displacement of Endpoint (DoE) (i.e., d at the endpoint of each repetition) of all the repetitions in the training data. The mean value $\mu_{TL}, \mu_{AoD}, \mu_{DoE}$ and standard deviation $\sigma_{TL}, \sigma_{AoD}, \sigma_{DoE}$ are calculated. According to the three-sigma rule [58], a detected repetition is an outlier and considered as extra repetition if

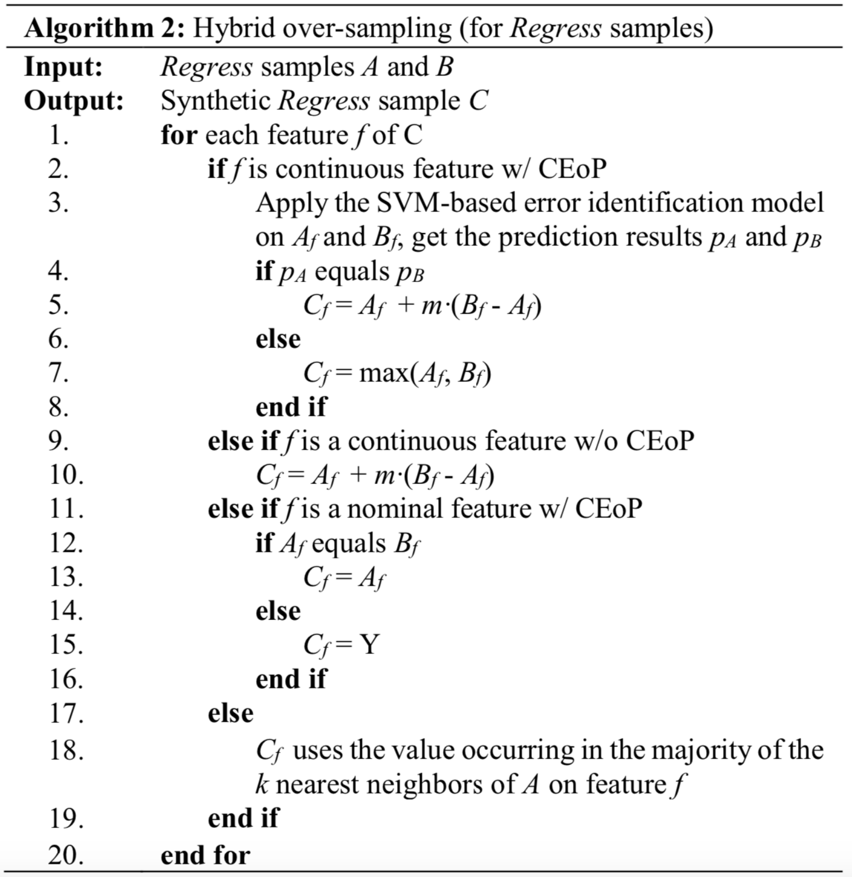
$$|TL - \mu_{TL}| > 3\sigma_{TL} \text{ or } |AoD - \mu_{AoD}| > 3\sigma_{AoD} \text{ or } |DoE - \mu_{DoE}| > 3\sigma_{DoE}. \qquad (3.4)$$

An extra repetition 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 for the following reasons. 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 on the training and test data will cause inaccuracy in state/sub-action segmentation. 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. To solve this problem, we propose using a second phase to enhance the accuracy in sub-action segmentation. First, we normalize the displacement data d of each repetition that is detected in the first phase. Second, the HMM-S model is applied on each repetition. Since the HMM-S model is a left-to-right model for single repetition, it is guaranteed that no extra repetitions will be detected. Therefore, sub-actions can be segmented based on the hidden state results in the second phase. Figure 3.7 shows the pseudo- code for the proposed TPHAU algorithm.

3.3.3 Movement Error Identification

In this section, we will introduce how to identify the patient's movement errors. For any task, the criteria used for evaluating the patient's performance have been defined by our PT co-author (see Table 3.3). 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 criterion). Based on the repetition detection and sub-action segmentation results, the patient's movement errors can be identified by checking the value of his/her 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 3.3), so we just need to check the knee angle (KA) of the back leg for frames in S_2 and S_3. The 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. 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. The patient's overall error on this criterion is calculated as the mean and maximum of e_{rep} for all the repetitions. The mean and maximum error e_{mean} and e_{max} will be used as features of the task recommendation model, which will be discussed in Section 3.3.4.

In addition to the quantitative errors, qualitative assessments (i.e., the patient's performance is either satisfactory or nonsatisfactory on a criterion) are also crucial in providing feedback for the patient. If the patient's performance on a criterion is nonsatisfactory, guidance will be rendered on the cloud and sent to the user's device to instruct him/her to improve the performance. Therefore, we build an SVM-based classification model [59]. For each training sample, the mean



```
Algorithm 2: Hybrid over-sampling (for Regress samples)
Input:   Regress samples $A$ and $B$
Output:  Synthetic Regress sample $C$
for each feature $f$ of C
    if $f$ is continuous feature w/ CEoP
        Apply the SVM-based error identification model
        on $A_f$ and $B_f$, get the prediction results $p_A$ and $p_B$
        if $p_A$ equals $p_B$
            $C_f = A_f + m \cdot (B_f - A_f)$
        else
            $C_f = \max(A_f, B_f)$
        end if
    else if $f$ is a continuous feature w/o CEoP
        $C_f = A_f + m \cdot (B_f - A_f)$
    else if $f$ is a nominal feature w/ CEoP
        if $A_f$ equals $B_f$
            $C_f = A_f$
        else
            $C_f = \mathrm{Y}$
        end if
    else
        $C_f$ uses the value occurring in the majority of the
        $k$ nearest neighbors of $A$ on feature $f$
    end if
end for
```

Figure 3.7: Pseudo-code of the proposed TPHAU algorithm.

and maximum errors on a criterion are used as the input feature. The label y of the sample is given by the PT based on the patient's performance during the data collection process, with $y = 1$ representing positive (i.e., the performance is satisfactory on this criterion) and $y = 0$ representing negative (i.e., the performance is non-satisfactory on this criterion). A linear binary SVM classifier is trained from the training data to find out the optimal decision boundary between the positive and negative samples. Since the criteria are independent of each other, a unique classification model is trained for each criterion of each task.

3.3.4 Machine Learning-Based Task Recommendation

In this section, we propose a task recommendation model to emulate the PT's decisions in updating the training tasks (i.e., the difficulty level for each task). Section 3.3.4 introduces the input and output of the model. Section 3.3.4 discusses the imbalanced data problem and existing methods. In Section 3.3.1, we propose a novel hybrid over-sampling approach to address the imbalanced data problem.

(1) Task Recommendation Framework

To enable automated task update recommendation, we propose a random forest-based classification model to emulate the PT's decision in updating the difficulty level of each task based on the patient's performance. Random forest (RF) is an ensemble learning method for classification, regression, and other problems [60]. Output of the proposed model is the PT's decision in updating the difficulty level, which are quantified into three categories: ***Progress*** (i.e., from level k to $k+1$), ***Repeat*** (i.e., repeat the current level k), and ***Regress*** (i.e., from level k to $k-1$). Note that a patient cannot progress any more when the current level is 4, but the PT may still assign progress if his/her performance is excellent in order to help the model learn the difference between ordinary and excellent performance. For the current level 1, the difference between *repeat* and *regress* are also clarified although outcomes for both situations are level 1. Inputs/Features of the model include the patient's maximum and mean error on each criterion (discussed in Section 3.3.3). Besides, some subjective factors (e.g., pain, age/sex, etc.) may also affect the PT's decision on task recommendation. For example, the PT may recommend *Repeat* to a patient with knee pain, even if the patient performs well on the current level. Table 3.4 shows all the features used in the task recommendation model.

Table 3.4: Features of the RF classifier.

Type	Feature	Value
Continuous	Maximum/mean error on a criterion	Criterion-specific
	Age	$56 \sim 89$
Nominal	Sex	M/F
	Current difficulty level	1/2/3/4
	Knee pain	Y/N
	Back/hip pain	

(2) Imbalanced Data Problem and Existing Methods

For the patient data that we have collected in the clinic, each sample (i.e., a patient performing a task once) belongs to one of the three categories (*Regress*, *Repeat*, *Progress*) based on the PT's recommendation on the task update. Table 3.5 shows the distribution of collected samples in the three classes for the three training tasks. We can see that the collected data are imbalanced for the three categories. The PT is conservative in regressing the patient, thus the percentage of samples in class *Regress* is very low (under 15%). As for *Repeat* and *Progress*, fewer patients (about only 20%) can progress to the next level for FL/BL than SQ. It may be because FL and BL are more challenging than SQ as they involve dynamic weight shift from on foot to the other, which is particularly difficult for patients with PD. Because of the imbalanced data problem, the RF classifier may be biased towards the majority class (e.g., class *Repeat* for FL) to achieve high overall accuracy. For example, a classifier applied on a training dataset with 95% positive samples and 5% negative samples can achieve high overall accuracy of 95% by using the simple strategy of always predicting positive. However, the cost of misclassifying a minority sample as a majority sample can sometimes be much higher than the cost of the reverse error. For example, predicting a patient who should *Regress* to the lower level (due to severe pain or errors) as *Repeat* and *Progress* may cause injury to the patient. Therefore, we should focus on the accuracy of each individual class instead of the overall accuracy. Next, we describe techniques that have been developed to address the imbalanced data problem in other applications, point out issues in utilizing these techniques, and subsequently propose a new technique for our application. Results of using our proposed technique in comparison with the existing methods will be provided in Section 3.4.4.

Table 3.5: Sample distribution for Squat (SQ), Forward Lunge (FL), Backward Lunge (BL).

Task	**Class (PT recommendation)**		
	Regress	*Repeat*	***Progress***
SQ	13.5%	42.5%	44.0%
FL	11.8%	67.8%	20.4%
BL	12.6%	63.2%	24.2%

Majority under-sampling [66]. It reduces the number of majority samples by selecting part of the majority samples. Because of the limited number of collected training samples in our task recommendation system, it may have negative effects on the accuracy.

Minority over-sampling with replacement [67]. It increases the number of minority samples by creating minority duplicates. However, Ling et al. [61] propose that it may cause over-fitting problem as it makes the decision region for the minority class more specific.

Decision threshold adjustment [65]. For a normal RF classifier, the probabilities of all the classes are calculated and the one with the highest probability is selected as final classification result. Provost et al. [65]] propose to tune the decision boundary to be biased towards the minority class, which is equivalent to assigning larger weight on the probability of the minority class. For the PT task recommendation problem, we can assign weights to the predicted probabilities as $w_{reg}P(Regress), w_{rep}P(Repeat), w_{prog}P(Progress)$ (w_{reg} may be greater than w_{rep} and w_{prog}) and then select the class with highest probability. However, Chawla et al. [62] has shown that simply changing the decision threshold cannot always guarantee better results.

Synthetic minority over-sampling [62]. The minority class is over-sampled by taking each minority sample and introducing synthetic samples between the sample and its nearest neighbors. The distance *dist* between two samples *A* and *B* is calculated as

$$dist = sqrt[\sum_{i=1}^{M}(A_{f_i} - B_{f_i})^2 + \sum_{j=M+1}^{M+N} \delta(A_{f_j}, B_{f_j}) Med^2] \tag{3.5}$$

$$\delta(A_{f_j}, B_{f_j}) = \begin{cases} 0, \text{ if } A_{f_i} = B_{f_i} \\ 1, \text{ otherwise} \end{cases} \tag{3.6}$$

where $\{f_1, \cdots, f_M\}$ are continuous features, $\{f_{M+1}, \cdots, f_{M+N}\}$ are nominal features, and Med is the median of standard deviations of all continuous features for the minority class. For continuous features, the Euclidean distance is included in dist. For nominal features, Med is included in $dist$ if A and B have different values on this feature. For each minority sample, k nearest neighbors are found and p neighbors among them ($p \leq k$) are randomly selected, depending on the over-sampling rate $p \times 100\%$. A synthetic sample is generated between the minority sample and each of the selected p neighbors. If p is not an integer, use $ceil(p)$ first and randomly select a percentage of $p/ceil(p) \times 100\%$ from all the synthetic samples. For continuous features f_c, linear interpolation is used to generate the new sample C as

$$C_{f_c} = A_{f_c} + m(B_{f_c} - A_{f_c}) \tag{3.7}$$

where m is a random number between 0 and 1. For nominal features, the value occurring in the majority of the k nearest neighbors is assigned to C. However, applying the traditional over-sampling method in our dataset does not give satisfying results. We will explain the problems and discuss the solutions in the next section.

(3) Proposed Hybrid Over-Sampling Approach

Based on the traditional synthetic minority over-sampling method [62], we propose a novel hybrid over-sampling approach. In this section, we will first introduce a pre-processing step (feature standardization), and then introduce the problems of applying the traditional over-sampling method [62] in our dataset and discuss our proposed solutions.

a) Feature standardization for continuous features

The continuous features we use in the task recommendation model (see Table 3.4) use different units of measurement and differ greatly in value range. For example, the value range of the patient's age is 56 $\sim$ 89 while the patient's error on the criterion "normalized length of step" has the value range of 0 $\sim$ 0.3. Therefore, features with greater values will dominate in the distance calculation in Equation (3.5) and features with smaller values may be ignored. To solve this problem, we propose a feature standardization step to preprocess the continuous features: all continuous features are normalized to zero-mean and unit variance before the distance calculation.

b) Hybrid interpolation for error features

There are some problems with the traditional linear interpolation approach Equation (3.7) when generating synthetic samples. We will first use the error features (i.e., patient's error on each criterion) as an example to illustrate the problem and propose our solutions, then generalize the solutions to the other features. Table 3.6 shows a simple example of the error features on two criteria C_1 and C_2. The PT recommends *Regress* for sample A and B for different reasons: A's performance on both criteria are non-satisfactory (Non-Sat) and B's error on C_2 is too large (50°). By using linear interpolation (with the random number m = 0.5 in Equation (3.7)), the synthetic sample C has error = 10° on C_1 (which may be Sat) and error = 30° on C_2 (Non-Sat). However, the PT may use complicated strategies in making recommendations instead of simply counting the number of Sat criteria. For example, if a sample has one Sat and one Non-Sat for the two criteria, the PT may recommend *Regress* only if the error of Non-Sat is too large (e.g., 50° on C_2 for sample B). Therefore, the PT may not recommend *Regress* for sample C since its error on C_2 is not so important. To create a correct *Regress* sample, we first propose a biased interpolation method based on the following fact: a *Regress* sample will still be in class *Regress* if any/all of its error features get larger in value. For the example in Table 3.6, a synthetic sample D that uses larger value of A and B on each error feature must also be a *Regress* sample.

Table 3.6: Different interpolation methods when generating synthetic samples.

Sample	**Error on C_1**	**Error on C_2**	**Recommendation**
A	20° (Non-Sat)	10° (Non-Sat)	*Regress*
B	0° (Sat)	50° (Non-Sat)	*Regress*
C (linear)	10° (Sat)	30° (Non-Sat)	*Regress*?
D (biased)	20° (Non-Sat)	50° (Non-Sat)	*Regress*
E (hybrid)	20° (Non-Sat, biased)	30° (Non-Sat, linear)	*Regress*

However, using biased interpolation on all the error features may cause the synthetic samples to be too far away from the original minority samples and the decision boundary to be not optimal for the original minority samples. Figure 3.8 shows an example of a majority class and a minority class. When the synthetic samples are far away from the original samples (see (a)), the decision boundary causes a high error rate on the original minority samples. To achieve the optimal over-sampling results, the synthetic samples should be among the original minority samples (as shown in (b)).

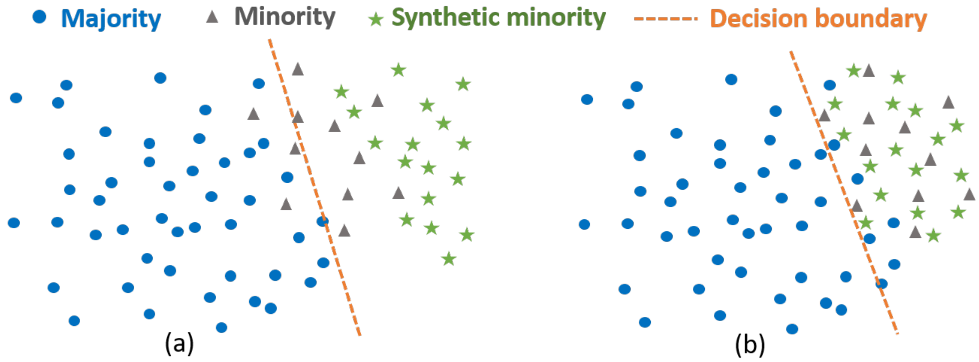


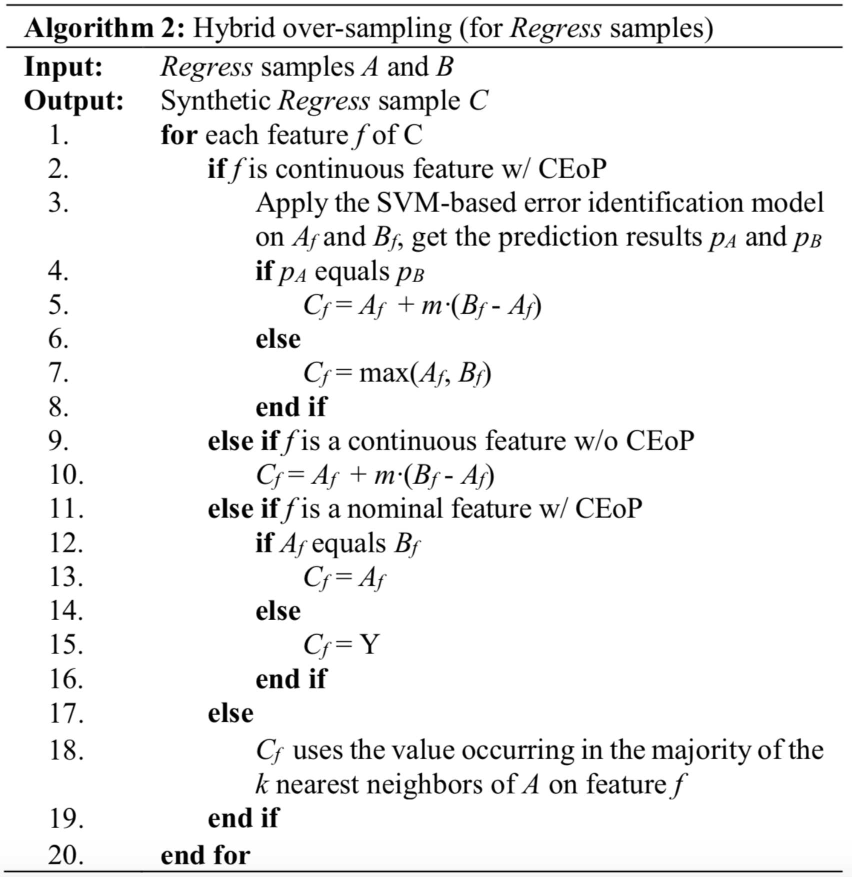
Figure 3.8: Minority over-sampling. (a) Synthetic samples are far away from original minority samples. (b) Synthetic samples are among original minority samples.

Therefore, we propose a hybrid interpolation approach to create synthetic over-sampling samples. When generating a synthetic *Regress* sample E from original *Regress* samples A and B, the SVM classifier (discussed in Section 3.3.3) is applied on A and B for each error feature. If both A and B are classified as Sat (or both are Non-Sat) on a criterion, linear interpolation in Equation (3.7) is used to create the value of the synthetic sample. Otherwise, biased interpolation will be used (i.e., use the larger error value of A and B). The last row in Table 3.6 illustrates this approach.

c) Hybrid interpolation: generalization to the other features

To generalize the proposed hybrid interpolation approach to the other features, we define features with Clear Effects on Performance (CEoP) as those features which can cause patient's better/worse performance, e.g., knee pain will cause worse performance. However, age and sex have no clear/direct effects on the performance. Based on the definition, we propose different interpolation methods for different features as follows. **(i) Continuous features w/ CEoP**: including patient's error on each criterion. Hybrid interpolation described in the previous section is used. **(ii) Continuous features w/o CEoP**: age. Since it has no clear effects on the performance, the proposed biased and hybrid interpolation approach cannot be used. Thus, we use linear interpolation on it. **(iii) Nominal features w/ CEoP**: including knee pain and back/hip pain. As linear interpolation (which is part of the proposed hybrid approach) cannot be used on nominal features, we use biased interpolation: if the two *Regress* samples differ in value (one is Y and the other is N), use Y for the synthetic *Regress* sample. **(iv) Nominal features w/o CEoP**: including sex and current difficulty level. Both biased and linear interpolation cannot be used on it. Hence, we use the value occurring in the majority of the k nearest neighbors for the synthetic *Regress* sample. The pseudo-code for the proposed hybrid interpolation approach is shown in Figure 3.9.

For class *Progress*, the same hybrid interpolation approach can be used for synthetic over-sampling except that a smaller error value and pain = N will be used in biased interpolation. For class *Repeat*, biased interpolation cannot be used since it is an intermediate class. From Table 3.5, we can see that class *Repeat* is not a minority class for all the three tasks discussed in this chapter, thus over-sampling is not needed for it.



```
Algorithm 2: Hybrid over-sampling (for Regress samples)
Input:   Regress samples $A$ and $B$
Output:  Synthetic Regress sample $C$
for each feature $f$ of C
    if $f$ is continuous feature w/ CEoP
        Apply the SVM-based error identification model
        on $A_f$ and $B_f$, get the prediction results $p_A$ and $p_B$
        if $p_A$ equals $p_B$
            $C_f = A_f + m \cdot (B_f - A_f)$
        else
            $C_f = \max(A_f, B_f)$
        end if
    else if $f$ is a continuous feature w/o CEoP
        $C_f = A_f + m \cdot (B_f - A_f)$
    else if $f$ is a nominal feature w/ CEoP
        if $A_f$ equals $B_f$
            $C_f = A_f$
        else
            $C_f = \mathrm{Y}$
        end if
    else
        $C_f$ uses the value occurring in the majority of the
        $k$ nearest neighbors of $A$ on feature $f$
    end if
end for
```

Figure 3.9: Pseudo-code of the hybrid over-sampling approach.

3.4 Results

In this Section, we will first introduce the data collection process, and then present the results of the proposed patient action understanding, error identification, and task recommendation models. We will also analyze and report the runtime efficiency of the proposed algorithms.

3.4.1 Experimental Setup and Data Collection

This research was approved by the Institutional Review Board at UC San Diego (protocol #181413X). 35 patients with PD (age 56 $\sim$ 89, 22 males, 13 females, Hoehn & Yahr stage 1 $\sim$ 4) recruited from the Neurological Rehabilitation Clinic, UC San Diego Health, participated in the data collection. All subjects signed the informed consent form. Each patient participated in the data collection for multiple times. Patient's motion data were recorded by a Microsoft Kinect v2 sensor. The corresponding PT assessments (i.e., whether the patient's performance was satisfactory or not on each criterion) and recommendations (i.e., *regress*, *repeat* or *progress*) were also recorded. For each task, the motion of one patient in one session constitutes a data sample. Each patient participated in the data collection for $2 \sim 4$ times. Note that sometimes some patients were not able to perform some tasks (e.g., BL was too difficult for some patients), thus the number of collected samples for each task were different. We collected 96 samples for SQ, 93 samples for FL, and 87 samples for BL in total. 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.

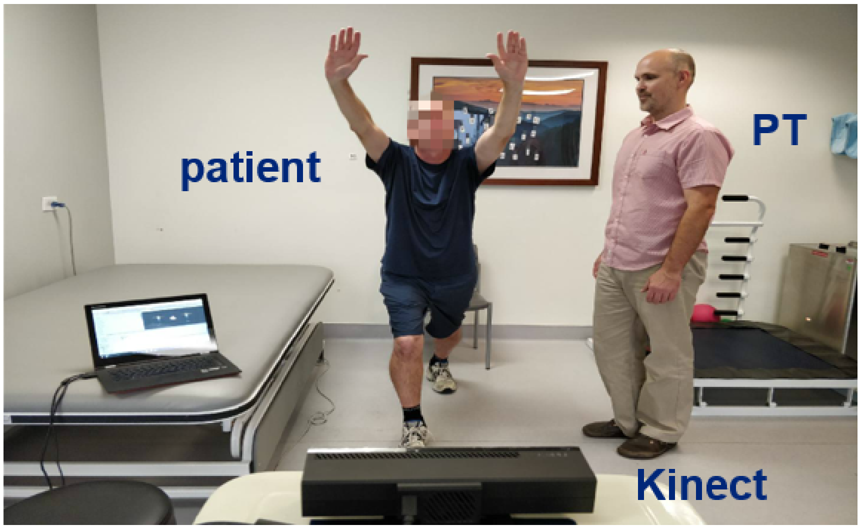


Figure 3.10: Data collection in PT clinic.

3.4.2 Patient Action Understanding Results

To validate the proposed TPHAU algorithm, we conduct experiments using the stratified 10-fold cross validation on SQ, FL, BL separately, with 90% of the samples for each task used for training and 10% for test. The comparison between the onephase Viterbi algorithm [57] and the proposed TPHAU algorithm is shown in Table 3.7. For repetition detection, the percentage of correct, wrong, missing repetitions, and the number of extra repetitions (discussed in Section 3.3.2) are calculated. We can see that the proposed TPHAU algorithm enhances the accuracy of repetition detection significantly, with more correct repetitions and much less extra repetitions, especially for BL. For sub-action segmentation, 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 sub-action $S_2/S_3/S_4$, the sensitivity and specificity are shown in Table 3.7. We can see that the proposed TPHAU algorithm improves both sensitivity and specificity for the three tasks. Especially for sensitivity, TPHAU enhances the sensitivity for S_3 significantly (e.g., from 78.3% to 94.4% for SQ). For S_2 and S_4, the average sensitivity of TPHAU is sometimes slightly lower than the one-phase Viterbi. For example, the sensitivity of S_2 in SQ is 89.5% using TPHAU, which is slightly lower than the sensitivity of 91.1% achieved by the one-phase Viterbi method. However, the small difference may be due to the PT's subjective bias when manually segmenting the states. Therefore, we can conclude that the overall accuracy of the proposed TPHAU algorithm outperforms the one-phase Viterbi method.

3.4.3 Patient Error Identification Results

To validate the SVM-based patient error identification method, we use the same training/test set as Section 3.4.2. A linear SVM classifier is trained for each criterion. The accuracy of each criterion is calculated as the ratio of the correctly classified samples to the total number of test samples. Table 3.8 shows the results for the three tasks. For most criteria, the accuracy is above 90%. For two criteria "Lean forward" in SQ and "Keep the front shank vertical" in BL, the accuracy is close to 90%. For only one criterion "Keep the back knee straight" in FL, the accuracy is 86.3%. Hence, it is reasonable to conclude that the SVM-based model can provide accurate error identification.

Table 3.7: Repetition detection and sub-action segmentation results for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

(a) Repetition detection results.

Method	**Task**	**Repetition detection**			
		Correct	**Wrong**	**Missing**	**No. of extra repetitions**
One-phase Viterbi [57]	SQ	90.6%	9.4%	0%	5
	FL	97.9%	2.1%	0%	14
	BL	96.3%	3.7%	0%	56
TPHAU (proposed)	SQ	97.1%	2.9%	0%	0
	FL	97.8%	2.1%	0%	2
	BL	99.4%	0.6%	0%	6

(b) Sub-action segmentation results.

Method	**Task**	**Sub-action segmentation**					
		Sensitivity			**Specificity**		
		S_2	S_3	S_4	S_2	S_3	S_4
One-phase Viterbi [57]	SQ	91.1%	78.3%	92.8%	93.6%	97.9%	94.2%
	FL	92.8%	86.2%	92.1%	96.2%	98.0%	94.1%
	BL	92.9%	81.0%	87.6%	92.1%	97.5%	96.7%
TPHAU (proposed)	SQ	89.5%	94.4%	90.8%	97.3%	98.1%	98.2%
	FL	93.5%	96.4%	92.5%	97.9%	98.2%	97.2%
	BL	92.0%	96.9%	88.4%	98.0%	97.5%	98.8%

Table 3.8: Accuracy of error identification models for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

Task	**Criterion**	**Accuracy**
SQ	Sit hips back towards a chair	92.5%
	Lean forward	89.1%
FL	Keep the back knee straight	86.3%
	Keep the posture tall	93.2%
	Length of step	93.8%
	Keep the front shank vertical	91.1%
BL	Keep the back knee straight	93.5%
	Keep the trunk parallel with the back leg	90.2%
	Length of step	94.2%
	Keep the front shank vertical	88.7%

3.4.4 Task Recommendation Results

To validate the proposed task recommendation approach, we build three RF-based task recommendation models for SQ, FL, BL separately. To solve the imbalanced data problem, we apply the techniques introduced in Section 3.3.4: majority under-sampling (***Under-Samp***) [66], minority over-sampling with replacement (***Over-Repl***) [67], traditional synthetic minority over-sampling using linear interpolation (***Over-Synth***) [62], decision threshold adjustment (***Thold-Adj***) [65], and the proposed hybrid synthetic over-sampling approach (***Proposed***). For under-sampling, the majority classes *Repeat* and *Progress* are under-sampled to a similar size of the minority class *Regress*. For over-sampling, class *Regress* is over-sampled to a similar size of class *Repeat*. Since class *Progress* also has less samples than class *Repeat* for FL/BL, we apply slight over-sampling on class *Progress*. (Between *Progress* and *Repeat*, a slight bias towards *Repeat* is preferred as the cost of misclassifying a *Progress* sample as *Repeat* is just delaying the patient's progress while the reverse error may cause health risks. Therefore, we apply slight instead of ordinary over-sampling on class *Progress*. Slight over-sampling means smaller over-sampling rate and fewer synthetic samples are created compared with ordinary over-sampling.) The accuracy of each class is calculated. Besides, the accuracy of class *Repeat* is affected by two types of errors: A) misclassifying *Repeat* as *Regress* (which may delay patient's progress), and B) misclassifying *Repeat* as *Progress* (which may cause risks). We consider the type B error more harmful, thus we also calculate the type B error as the False Positive Rate (FPR) of class *Repeat*. The original results (without using any method to solve the imbalanced data problem) and results by using these techniques are shown in Table 3.9. For the original imbalanced dataset, the majority class (i.e., *Repeat* and *Progress* for SQ, *Repeat* for FL and BL) achieves high accuracy (around 90%) while the accuracy of the minority class *Regress* is much lower (below 70%). Among all the methods, ***Over-Repl*** and *Over-Synth* are not able to improve the accuracy of class *Regress* significantly. The two methods *Under-Samp* and *Thold-Adj* increase the accuracy of class *Regress*, however with the cost of high FPR of class *Repeat* (e.g., FPR of *Repeat* is 9.5% using *Under-Samp* and 7.9% using

Thold-Adj for FL). Overall, our proposed hybrid synthetic over-sampling approach outperforms the other methods in increasing the accuracy of the minority class while maintaining high accuracy and low FPR on the majority class. To show the importance of including the subjective factors (discussed in Section 3.3) in the PT's recommendation, we conduct experiments by removing all the subjective factors from the features and applying the proposed hybrid over-sampling approach. Results are shown in the last row of Table 3.9. We can see that accuracy drops significantly, especially for class *Regress*. It is reasonable since some of the subjective factors (e.g., knee pain) indicate patient's poor health condition, which may be the primary reason of PT's decision to regress the patient.

Table 3.9: Accuracy of error identification models for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

Task	**Method**	**Accuracy**			FPR (***Repeat***)
		Regress	***Repeat***	***Progress***	
SQ	*Original*	69.2%	91.7%	95.7%	2.8%
	Under-Samp [66]	84.6%	75.0%	87.2%	11.1%
	Over-Repl [67]	76.9%	88.9%	95.7%	11.1%
	Over-Synth [62]	69.2%	88.3%	95.7%	11.1%
	Thold-Adj [65]	92.3%	86.1%	91.5%	5.6%
	Proposed	92.3%	88.9%	95.7%	2.8%
	No Subjective factors	76.9%	86.1%	95.7%	11.1%
FL	*Original*	54.5%	88.1%	76.5%	3.2%
	Under-Samp [66]	81.8%	81.0%	73.7%	9.5%
	Over-Repl [67]	63.6%	85.5%	75.0%	9.7%
	Over-Synth [62]	72.7%	83.9%	85.0%	8.1%
	Thold-Adj [65]	81.8%	68.3%	84.2%	7.9%
	Proposed	81.8%	85.7%	84.2%	3.2%
	No Subjective factors	72.7%	85.7%	84.2%	6.3%
BL	*Original*	63.6%	92.7%	85.0%	3.6%
	Under-Samp [66]	81.8%	67.2%	85.0%	7.3%
	Over-Repl [67]	63.6%	92.7%	85.0%	3.6%
	Over-Synth [62]	72.7%	89.1%	90.0%	3.6%
	Thold-Adj [65]	81.8%	72.7%	90.0%	7.3%
	Proposed	81.8%	90.9%	90.0%	5.4%
	No Subjective factors	27.3%	76.4%	90.0%	7.3%

3.4.5 Running Efficiency of the Proposed Algorithm

For the three algorithms proposed in this chapter, we have conducted comprehensive experiments to test their runtime efficiency. The running time of each algorithm is tested on an Intel Xeon E5-1650 CPU. For each algorithm, there is an offline training stage (in which the model is trained on training samples) and a test/inference stage (in which the trained model is applied on new/test samples). Since the proposed virtual PT system is cloud-based, high efficiency is needed in the inference stage to provide action understanding, error identification, and task recommendation in a timely manner. Therefore, the runtime efficiency in the inference stage is of greater importance. In this section, we will present the running time of each algorithm in both training and inference stages.

(1) The TPHAU algorithm

It is compared with the one-phase Viterbi algorithm. The running time of the training and inference stage is shown in Table 3.10. The total training time is the total time taken to train the model on all the training samples. The average inference time is the average running time of applying the model on a new/test sample. Since the two algorithms differ only in the reference stage, their training time is the same (about 20 s for each task). In the inference stage, the proposed TPHAU algorithm requires more running time due to the use of the second phase to improve the detection accuracy (discussed in Section 3.3.2). From Table 3.10 we can see that it takes less than 150 ms to apply the proposed TPHAU algorithm on a new/test sample in the inference stage, which means that action understanding can be performed in real time.

(2) The error identification model

For error identification, a SVM classifier is used to identify whether the patient's performance is satisfactory or not on a PT-defined criterion. Since multiple criteria have been defined

Table 3.10: Running time of the one-phase Viterbi algorithm and the proposed TPHAU algorithm, for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

Method	**Total training time (s)**			**Average Inference Time (ms)**		
	SQ	**FL**	**BL**	**SQ**	**FL**	**BL**
One-phase Viterbi [57]	17.3	21.3	20.2	65.4	102.2	111.8
Proposed TPHAU				81.9	127.5	139.5

for each task (discussed in Section 3.3.1), the running time of each criterion is summed up as the total running time needed to evaluate the patient's performance on all PT-defined criteria for this task. We summarize the running time in both training and inference stage in Table 3.11. We can see that the training stage requires less than 30 ms for each task. The inference stage is very fast, requiring less than 0.1 ms for each task.

Table 3.11: Running time of the proposed error identification model, for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

Method	**Total training time (s)**			**Average Inference Time (ms)**		
	SQ	**FL**	**BL**	**SQ**	**FL**	**BL**
Proposed SVM model	14.9	27.4	27.1	0.02	0.04	0.05

(3) The task recommendation model

The proposed task recommendation model is based on the random forest classifier. Because of the imbalanced data problem (discussed in Section 3.3.4), we have proposed the hybrid synthetic over-sampling approach to generate synthetic samples for the minority class in the training stage and have shown its results compared with other methods (discussed in Section 3.3.4 and Section 3.4.4). Table 3.12 shows the total training time required by each method for the imbalanced data problem. We can see that the training time of the traditional synthetic over-sampling approach (Over-Synth) and the proposed hybrid synthetic over-sampling approach (Proposed) is higher than the other techniques because these two methods requires extra steps to generate the new synthetic samples. For the inference stage, the average inference time is the same for all the methods since these methods are applied only in the training stage to address the imbalanced data problem. We can see that the inference stage of the task recommendation model requires only 4 ms for each task.

Table 3.12: Running time of the proposed error identification model, for Squat (SQ), Forward Lunge (FL), and Backward Lunge (BL).

Method	**Total training time (s)**			**Average Inference Time (ms)**		
	SQ	**FL**	**BL**	**SQ**	**FL**	**BL**
Original	2.9	3.2	3.1	3.9	4.0	4.1
Under-Samp [66]	2.8	3.0	2.8			
Over-Repl [67]	2.9	3.2	3.0			
Over-Synth [62]	6.1	13.5	11.2			
Thold-Adj [65]	2.9	3.1	3.1			
Proposed	6.6	15.9	14.0			
No Subjective factors	5.1	10.0	9.0			

From the results presented above, we can see that the running time of the three proposed models (i.e., the TPHAU algorithm for patient action understanding, the SVM-based error identification model, and the task recommendation model) in the inference stage is about 150 ms in total. It means that the virtual PT system can evaluate the patient's performance and provide task recommendation in about 150 ms after the patient completes a training task, which enables efficient and real-time remote care.

3.5 Conclusion

In this chapter, we propose a virtual PT system to enable ondemand remote training for patients with PD. Patient's movements can be understood by the proposed TPHAU algorithm and errors are identified by SVM-based models. To enable automated task recommendation, a machine learning-based model is developed and trained from real patient data, which can emulate the human PT's recommendations. Experiments on patient data show that the proposed methods can accurately understand the patient's actions, identify errors, and provide task recommendation like a real PT. The proposed virtual PT system has the potential of enabling on-demand virtual care and significantly reducing cost for both the patients and care providers.

In Chapter 2 and Chapter 3, we have introduced the virtual PT system, which can provide remote training, assessment and task recommendations for patients who need physical therapy. However, an important aspect, i.e., balance evaluation, was absent in the training system. Therefore, in Chapter 4, we will propose an automated balance evaluation system using multiple sensors to enable quantitative balance evaluation for patients with balance problems.

Chapter 3, in part, is from the material as it appears in proceedings of IEEE International Conference on Healthcare Informatics 2018. Wenchuan Wei; Carter McElroy; Sujit Dey. and in IEEE Transactions on Neural Systems & Rehabilitation Engineering 2019. Wenchuan Wei; Carter McElroy; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

Chapter 4

Using Sensors and Deep Learning to Enable On-Demand Balance Evaluation

4.1 Introduction

In physical therapy, the patient's ability to balance is an important indicator for the physical therapist (PT) to select the proper training programs, evaluate the progress of the patient, predict fall risk [68], etc. Traditionally, balance evaluation is performed by the PT at the initial evaluation and intermittently during clinic visits. However, the patient's balance may change over time and also be influenced by medication, sleep quality, etc. Therefore, it is important to have more frequent and preferably on-demand balance evaluation to monitor the patient's condition. Moreover, traditional balance evaluation tests like the Berg Balance Scale (BBS) [69] and the mini Balance Evaluation Systems Test (mini-BESTest) [70] are time-consuming and require the PT's subjective assessments, therefore they may be limited for clinical use. To address the problems of traditional balance evaluation, Mishra et al. have proposed to use a camera system to evaluate the static balance (i.e., the ability to stay stationary in some postures) using static body sway in single-leg stance [71]. For dynamic balance (i.e., the ability to maintain balance in motion or recover from imbalanced conditions), Kennedy et al. have proposed the WeHab system to measure the patient's balance in dynamic tasks (e.g., sit-to-stand and weight-shifting) but do not achieve good results [72]. In this chapter, we focus on the dynamic balance evaluation for patients with Parkinson's disease (PD) as dynamic balance is more important to improve agility and avoid falls. We propose an automated balance evaluation system using multiple sensors and deep learning to provide accurate, convenient, and on-demand balance evaluation for home and clinical use.

In balance evaluation, an important indicator is the Center of Mass (CoM) position of the human body. For the 3D position of the human's CoM, the horizontal CoM (i.e., the projection of CoM on the ground) is of greater importance [71, 72]. Since the CoM position of the human body cannot be directly measured, researchers have proposed to measure the Center of Pressure (CoP) of the ground reaction force in static/balanced postures to represent the horizontal CoM position [77, 78, 79, 87]. In a static/balanced posture (e.g., quiet standing), the only forces acting on the human body are the gravity (which acts on the CoM) and the ground reaction force. According to Newton's second law, the gravity is equivalent to the ground reaction force in both magnitude and position (i.e., CoP = horizontal CoM) since the acceleration of the human body is zero in static/balanced postures. The traditional way to measure the CoP position is using the laboratory-grade force plate. However, the force plate is primarily limited to laboratory use due to its high cost and complicated setup procedure. The Wii Balance Board (WBB) is a device designed by Nintendo for balance-related games and can calculate the CoP position of the human body. The CoP measurement error of the WBB has been proved to be within 5 *mm* [73]. Because of its low cost, portability, and high accuracy in CoP measurement, the WBB has been increasingly used as a replacement of the force plate in many studies [73, 74, 75].

However, the CoP position measured by the force plate or the WBB is equivalent to the horizontal CoM position only when the user is in a static/balanced posture. Moreover, the force plate or the WBB needs to be placed on a horizontal and firm plane to measure the CoP position accurately. In balance evaluation, we often need to test the subject's dynamic balance or the subject's static balance on different surface types (e.g., the incline ramp, or the foam). To solve this problem, researchers have proposed to use pose and body parameters (e.g., body shape and density) to estimate the CoM position. In previous studies, the body shape of the subject is either modeled as geometrical segments [76, 77] or estimated from an identification/calibration process [78, 87]. To achieve identification-free CoM estimation, Kaichi et al. have proposed a voxel reconstruction approach to reconstruct the subject's 3D body using multiple cameras and estimate the CoM position by assigning weights to the body parts [79]. However, they need to carefully calibrate five cameras for the 3D body reconstruction, which makes it not suitable for home and clinical use.

In recent years, vision-based models have been increasingly used to learn and predict human-related activities, for example, facial expression recognition [80], fall prediction [81], etc. Inspired by these techniques, we propose to use deep learning to learn the body parameters of the subject and estimate the horizontal CoM position. We have selected the depth camera instead of an RGB camera because the depth map provides more information about the subject's body in the depth direction, which is essential in CoM estimation. Besides, depth cameras work better in low light conditions and are color and texture invariant [82]. Figure 4.1 shows the proposed CoM estimation model. Motivated by the use of Convolutional Neural Network (CNN) in pose estimation problems [102, 103], we propose to use CNN in our CoM estimation model as estimating the human's CoM position is similar to estimating the joint positions (i.e., pose estimation). In the training phase, a CNN-based model is trained using data collected from multiple subjects in various static postures. We use a depth camera to capture the depth images and a WBB to measure the ground-truth CoP position. In the application phase, only the depth camera is needed to estimate the subject-specific CoM position. The depth camera is anyway necessary in most automated training systems for its ability in skeleton tracking and motion capture [83, 84]. By using the proposed CoM estimation model, the subject's CoM position can also be tracked without any extra device.

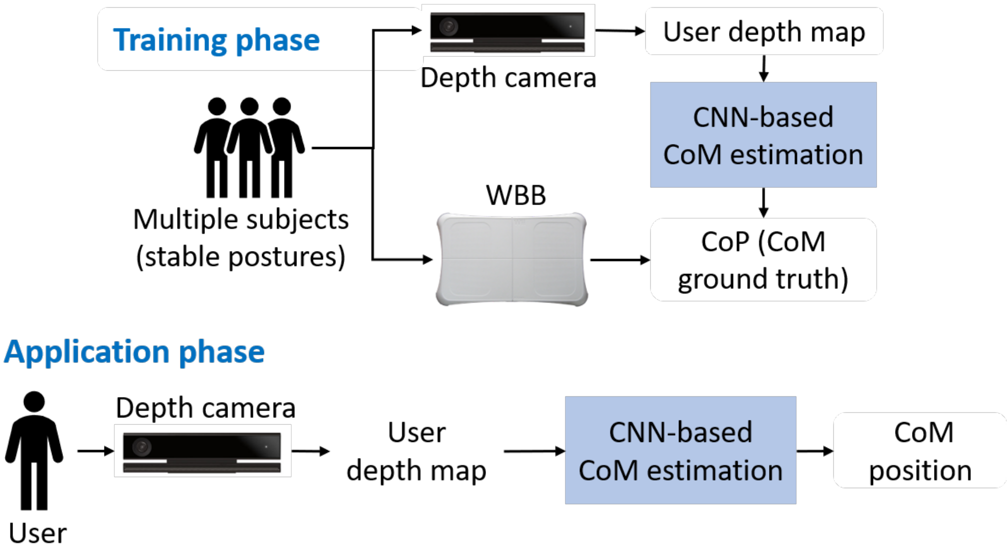


Figure 4.1: The training and application phase of the proposed CoM estimation model.

Note that the CoM estimation model is trained from data collected in static postures and it will be used for dynamic postures in the balance evaluation system. Despite the fact that there is no direct way to validate its accuracy on dynamic postures (as the ground truth of CoM position cannot be measured), we will demonstrate that the balance evaluation model built upon the CoM estimation model is able to provide accurate balance assessments that are consistent with the PT score. Therefore, it is reasonable to conclude that the proposed CoM estimation model can provide accurate CoM estimation for both static and dynamic postures. By using a single depth camera that does not need complicated setup or subject identification, the proposed CoM estimation model can be used as a portable and low-cost tool for subject-specific CoM measurements.

Based on the CoM estimation model, we further propose the balance evaluation system using multiple sensors. The tested task is Gait Initiation (GI), which refers to the transient period between the quiet standing posture and steady state walking. Patients with impaired balance have difficulty in performing the correct body weight shift in GI [97]. Hass et al. have proposed that the CoP-CoM distance during GI is an important indicator of dynamic balance control [85]. Inspired by their research, we propose to develop an automated balance evaluation system to provide quantitative balance evaluation using the GI task and mimic the human PT's assessments during traditional balance tests. The proposed system is shown in Figure 4.2. The depth camera and the WBB measures the subject's CoM and CoP positions during the GI task respectively. The patient's balance level will be calculated based on the CoP and CoM trajectory. To the best of our knowledge, our proposed system is the first to provide automated and quantitative evaluation on the subject's dynamic balance, which can mimic the human PT's manual assessments in the mini-BESTest. While we focus on patients with Parkinson's disease (PD) in this chapter, the proposed balance evaluation system can be used in the physical therapy for any disease/condition where balance evaluation is critical (e.g., orthopedic disease and stroke).

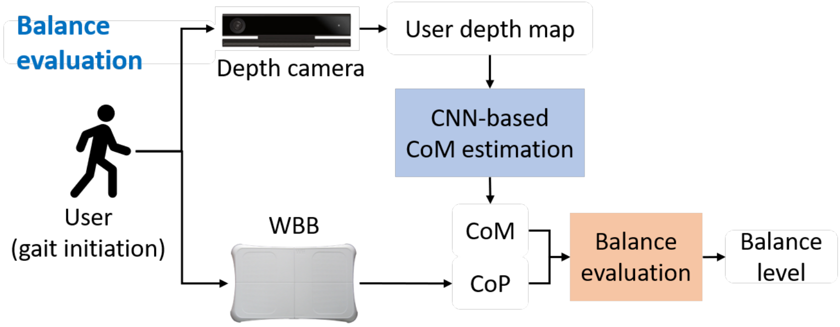


Figure 4.2: The proposed balance evaluation system.

A preliminary version of this work has been reported in [86], which introduced a CoM estimation model. However, the model proposed in [86] did not show high accuracy. In this dissertation, we develop an enhanced CoM estimation model by using colored skeleton images instead of joint heatmaps as inputs to the model, and proposing a novel coarse-to-fine approach to improve the accuracy. The enhanced model reduces the estimation error by about 10%, compared with the preliminary model in [86]. Moreover, our preliminary work [86] proposed only the CoM estimation model, whereas this work uses the enhanced CoM estimation model to propose an automated balance evaluation system, which for the first time enables quantitative, accurate, and on-demand dynamic balance evaluation for home and clinical use. Compared with traditional balance evaluation (e.g., the mini-BESTest conducted by a PT, or tests using laboratory-grade devices), the proposed balance evaluation system can be used at home or away (e.g. at hotels while traveling), or in clinics without PTs or high-end devices (e.g., retail-based clinics and mobile clinics). The patients can use the proposed system as a portable and low-cost tool to measure their balance on an on-demand basis, which enables closer monitoring of their health condition and progress in physical therapy training. The proposed balance evaluation system has the potential of significantly reducing PT visit requirements and reducing cost for both the patients and care providers.

The rest of the chapter is organized as follows: Section 4.2 introduces the related work on CoM estimation and balance evaluation in more details. In Section 4.3, we introduce the methods used in the proposed models, including the CoM estimation model in Section 4.3.2 and the balance evaluation system in Section 4.3.3. Section 4.4 describes the experimental results. Section 4.5 concludes this chapter and discusses future work.

4.2 Related Work

While we have briefly discussed the related work on CoM estimation and balance evaluation in the previous section, we next explain the most relevant techniques in more details, pointing out their disadvantages and the need and differentiation of our proposed technique.

4.2.1 Related Work on CoM Estimation

CoM estimation using IMU sensors [100, 101]: Some studies used Inertial Measurement Unit (IMU) sensors to estimate the CoM position. Esser et al. proposed to estimate the subject's vertical CoM movements from the acceleration data collected by the IMU sensor by [100]. However, the wearable IMU sensors are not convenient for patients with impaired mobility.

Winter's method: Winter proposed a kinematic method to estimate the CoM position of the human body [76]. He modeled the human body as 16 segments and used a motion capture system to track the position of each segment. The CoM position of the whole body was calculated as the weighted sum of the CoM position of each segment. The weight of each segment was taken from previous anthropometric studies. However, this method cannot provide subject-specific CoM estimation as the weight of each segment may differ in subjects of different age, sex, and fitness level, etc.

The optimization-based method: Chen et al. proposed to use an optimization-based model to estimate the body parameters of the subject [77]. They modeled the human body as some geometric shapes and measured the size of each segment manually. A force plate was used to measure the CoP position as the ground truth of the horizontal CoM position. However, modeling the body segments as geometrical shapes (e.g., modeling the neck as a frustum) is not accurate and the manual measurement of the body size is inconvenient.

The Statically Equivalent Serial Chain (SESC) model: The SESC model translates the human's mass distribution to the geometry of a linked chain [78]. An identification phase was used to obtain the subject-specific SESC parameters. In the identification phase, each subject performed 14 static postures. Later, Gonzalez et al. proposed that using more postures in the identification phase and assuming the bilateral symmetry of the human body can reduce the estimation error of the SESC method [87]. They also showed that using low-cost sensors Kinect and WBB can achieve comparable results to those obtained using high-end equipment. However, the subject identification phase still needs to be conducted each time when a new subject comes or the mass distribution of an existing subject has changed, which limits its application.

The voxel reconstruction method: Kaichi et al. proposed to reconstruct the 3D human body and then estimate the CoM position [79]. They used five cameras to capture multiple views of the human body and a 3D reconstruction approach to reconstruct the body. The human body was segmented into nine parts and the CoM position of the whole body was estimated as the weighted sum of the position of each part. The weights were taken from previous anthropometric studies. As mentioned in Section 4.1, the main challenges in the subject-specific CoM estimation problem include the difference in body size and density. By reconstructing the 3D body, the voxel reconstruction approach solves the problem of difference in body size but still fails to consider the difference in body density since it uses the density information from previous studies. Moreover, the five cameras need to be carefully calibrated. In comparison, our proposed model uses a single depth camera and does not need any complicated calibration or subject identification process, which is more convenient for home and clinical use.

4.2.2 Related Work on Balance Evaluation

The balance control of the human body includes static balance and dynamic balance. Static balance refers to the ability to stay stationary in some postures (e.g., single-leg stance), while dynamic balance refers to the ability to maintain balance in motion or recover from imbalanced conditions. For static balance evaluation, the body sway during single or two-legged stance is used. The body sway is presented by the moving range of the CoM positions, which can be measured by a force plate or a WBB (as CoP = CoM in static conditions) [88], or estimated using the above CoM estimation methods [71]. Subjects with better static balance would have smaller body sway. For dynamic balance, Hsu et al. proposed to use an inertial-sensor-based wearable device to analyze gait information and balance ability for patients with Alzheimer's disease [99]. However, wearable sensors attached on the body may cause extra burden to the users, especially for patients with impaired mobility. Therefore, we decide to use non-wearable sensors (e.g., cameras and balance boards) in the proposed balance evaluation system for patients with PD. Hass et al. proposed that the CoP-CoM distance during the GI task might represent the dynamic balance control of patients with PD and shown that the peak magnitude of the CoP-CoM distance was smaller in more balance-impaired patients than in healthy subjects. However, the CoM measurements in their work were based on the skeleton-based approach [76] and were not accurate. Moreover, they provided only qualitative results by showing the difference in CoP-CoM distance between patients with PD and healthy subjects. In comparison, our proposed balance evaluation model is able to provide quantitative balance level, which is consistent with the human PT's manual assessments in standardized balance tests. The quantitative balance level can be used to select the proper training programs, evaluate the patient's progress, and predict the fall risk.

4.3 Methods

4.3.1 Devices: Kinect and Wii Balance Board

The Kinect sensor can capture the human pose using an RGB camera and a depth camera [89]. Each pixel in the depth map represents the distance of the pixel from the sensor. Based on the original depth map, the user depth map (by removing the background) and the user skeleton can be obtained [82] (see Figure 4.3).

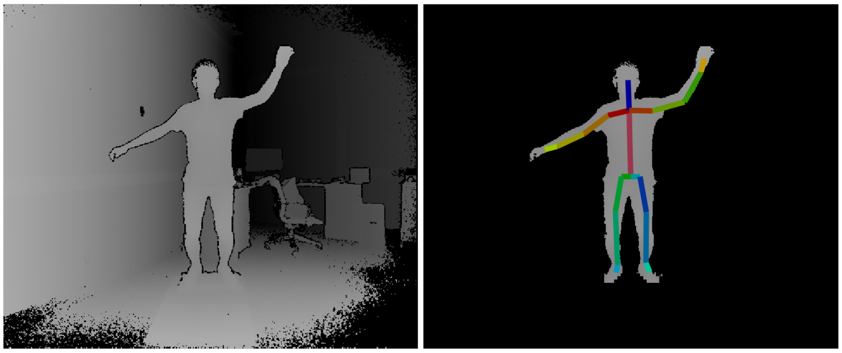


Figure 4.3: Left: the original depth map captured by the depth camera. Right: the user depth map and the colored skeleton image overlay.

The Wii balance board (WBB) consists of four pressure sensors located at the four corners of the board. When a user stands on the board, the four pressure sensors measure the vertical force and the CoP can be calculated. Compared with our preliminary work in [86], we have extended the range of CoP measurements by using two WBBs side by side to enable more postures. Figure

4.4 shows the two WBBs and the coordinate system. In this chapter, the x- and y-axis are defined as the length and width direction of the WBB, and the z-axis is the upright direction. Based on torque equilibrium, the CoP position can be calculated as

$$x = \frac{L}{2} \times \frac{(P_{12}+P_{14}+P_{22}+P_{24})-(P_{11}+P_{13}+P_{21}+P_{23})}{P_{11}+P_{12}+P_{13}+P_{14}+P_{21}+P_{22}+P_{23}+P_{24}}, \tag{4.1}$$

$$y = \frac{(t+W)(P_{11}+P_{12}-P_{23}-P_{24})+t(P_{13}+P_{14}-P_{21}-P_{22})}{P_{11}+P_{12}+P_{13}+P_{14}+P_{21}+P_{22}+P_{23}+P_{24}}, \tag{4.2}$$

where L and W are the length and width of the board, t is the size of the gap between the two boards, and P_{ij} is the force measured by the j-th pressure sensor of the i-th board. Several studies have found that the CoP measurement error of the WBB is smaller than 5 mm, compared with the laboratory-grade force plate [73, 90]. Besides, the WBB is inexpensive and portable, which makes it a good tool for home and clinical use. Therefore, we have selected the WBB to measure the CoP positions in this work.

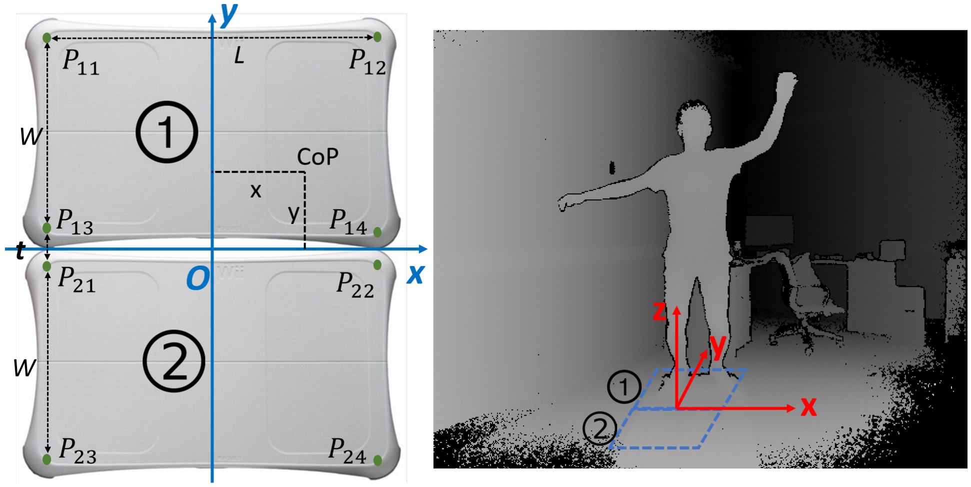


Figure 4.4: Two WBBs and the 3D coordinate system.

4.3.2 The Proposed CoM Estimation Model

(1) Input and Output of the Model

For the CoM estimation model, the input is the full depth map and the output is the horizontal CoM position of the user. To help the model distinguish between different body parts (as different parts may have different densities), we have proposed in our preliminary work [86] to use the joint heatmaps to provide information about the joint positions. However, the joint heatmaps are high-dimensional and introduce too many parameters in the CNN model. As the heatmap of each joint has the same size as the input depth image (512×424), the heatmaps of all 25 joints have 25 channels ($512 \times 424 \times 25$). To reduce the number of parameters in the CNN model, we further propose to use the colored skeleton image instead of the joint heatmaps to provide information about the different body parts of the subject. The colored skeleton image is created by connecting the adjacent joints of the body and using a specific color for each body segment. For example, the right shank connecting the right knee joint and the right ankle joint is rendered in light blue (RGB = [0, 102, 153]). Figure 4.3 shows an example of the colored skeleton. The colored skeleton image also has the same size as the depth image (512×424) but has only 3 channels, compared with the 25 channels of the joint heatmaps proposed in [86]. Therefore, the colored skeleton image can reduce the training and inference times of the CNN model by reducing the input dimension and the number of parameters of the model. Each body segment is rendered in a different color so the network can differentiate between different body parts. The user depth map and the colored skeleton image are concatenated as the input of the CNN model.

The output of the model is the horizontal CoM position of the user. As shown in (4.1) and (4.2), the horizontal CoM positions measured by the WBB are continuous values (x, y), therefore the CoM estimation is a regression problem. However, it has been proved that the direct regression of coordinates from images is a highly non-linear problem and learning the

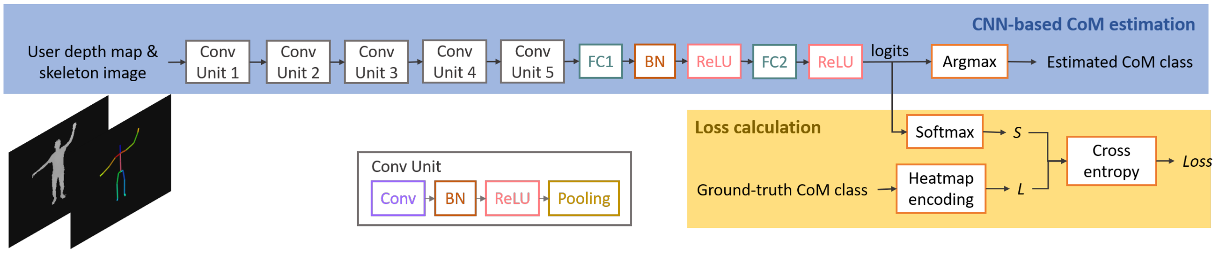


Figure 4.5: The proposed CNN model for CoM estimation.

mapping is a challenging task [91]. To solve this problem, we propose to discretize the continuous coordinates into discrete classes. For each data sample, the CNN model will predict the most likely discretized class k_x and k_y ($k_x, k_y = 0,\ 1,\ 2,\ \ldots$) and the continuous CoM coordinate will be estimated as the center of the discretized class as

$$CoM_x = (k_x + 0.5) \times I_x, CoM_y = (k_y + 0.5) \times I_y \tag{4.3}$$

where I_x and I_y are the length of the discretization interval (DI) in the x- and y- direction. More details about the selection of DI will be discussed in Section 4.3.2. By discretizing the continuous CoM coordinates, we cast the highly non-linear problem of direct CoM coordinate regression to a more manageable form of classification in a discretized space.

(2) Data Augmentation

Data augmentation is an important step in deep learning to increase the amount and diversity of the training data and reduce overfitting. Traditional data augmentation approaches include rotating, flipping, translating the image, and/or adding noise to the image. In image classification, these operations are useful as they do not change the image categories. However, they cannot be directly applied to our dataset as the CoM position of the user may be different. To solve this problem, we propose to apply different data augmentation approaches to the x- and y-component of the CoM position separately.

For the *x*-component of the CoM position, two data augmentation approaches are applied to the user depth map: (1) Adding a random depth value to the user body area, which is identical to shifting the user body in the depth direction. (2) Shifting the user body randomly in the *z*-direction. Both operations will not change the *x*-value of the CoM position. For the *y*-component of the CoM position, two data augmentation approaches are applied to the user depth map: (1) Shifting the user body in the *x*-direction randomly. (2) Shifting the user body in the *z*-direction randomly. Both operations will not change the *y*-value of the CoM position. Note that the colored skeleton images also need to be processed in the same way as the user body (i.e., adding the same depth value and shifting the same amount).

(3) CNN-based Network Architecture

In computer vision problems, CNN [92] is widely used for its advantages in feature extraction, parameter sharing, etc. We propose a CNN-based model for the CoM estimation problem (see Figure 4.5). In each convolutional unit, we use a Convolutional (Conv) layer [93] to extract features from the original image or the output of the previous layer, a Batch Normalization (BN) layer [94] to stabilize the inputs to the following nonlinear activation function, a Rectified Linear Unit (ReLU) layer to add non-linear transformation, and a max Pooling layer to reduce the size of each feature map. We use five Conv units to extract features from the depth images. The number of layers is selected empirically and details about our implementation are shown in Section 4.4.2. After the five Conv units, we use two Fully Connected (FC) layers to output the probability of each discrete CoM class from the results of previous Conv units, and an Argmax layer to select the final output with the highest probability. As described in Section 4.3.2, the continuous CoM positions have been discretized into some classes, so the CNN model will do a classification to decide the correct class of the CoM coordinates. We define the loss function as the cross-entropy between the ground-truth class of the CoM and the predicted CoM class as follows.

$$Loss = -\sum_{i=1}^{N} L_i log(S_i), \tag{4.4}$$

where L_i is the encoding for class i in the ground-truth CoM and S_i is the softmax output of class i in the estimated CoM. In most image classification problems, the traditional encoding method for the ground-truth label is one-hot encoding as follows.

$$L_i = \begin{cases} 1, i = k \\ 0, i \neq k \end{cases} \tag{4.5}$$

where k is the ground-truth class. In this way, the ground-truth class k is encoded as 1 and all the other classes are encoded as 0. Figure 4.6 shows an example. One-hot encoding is used in image classification problems because the label for an image is a categorical feature and all the incorrect classes ($i \neq k$) should be considered equally. However, the ground-truth class of CoM position is discretized from the continuous value, so the incorrect classes should be penalized differently according to their distance to the ground-truth class. Thus, we propose to use Gaussian-distributed heatmap instead of one-hot encoding to encode the ground-truth CoM as

$$L_i = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(i-k)^2}{2\sigma^2}}, \tag{4.6}$$

where σ is the standard deviation of the Gaussian distribution. An example of the Gaussian heatmap is also shown in Figure 4.6. The ground-truth class k has the highest probability 0.20 and the other classes are encoded according to their distance to the ground-truth class k. The CoM heatmap represents the confidence of each class as the ground truth. By using the Gaussian heatmap, the CNN model can be trained to move its output towards the ground-truth class during the learning process.

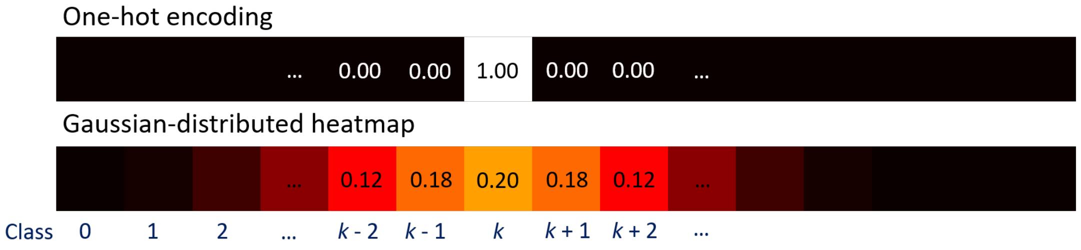


Figure 4.6: CoM ground-truth class encoding (*k* is the true class): one-hot encoding and Gaussian-distributed heatmap.

(4) A Coarse-to-Fine Approach to Increase the Accuracy

As discussed in Section 4.3.2, the continuous CoM coordinates are discretized into some classes in the CNN model. However, there are some trade-offs in the selection of the discretization interval (DI) when discretizing the CoM coordinates. Smaller DI leads to larger number of discretized classes and therefore more challenges in the classification problem due to some outliers. Figure 4.7 shows an example. The numbers in each block represent the output probability of each class. The outlier class has a probability 0.16, which is higher than the correct class (probability = 0.15). For larger DI, there are smaller number of classes, which leads to higher accuracy in the classification problem. However, the final CoM estimation error may still be high as the true CoM position within the class may be far from the center of the interval that is estimated as the output CoP position.

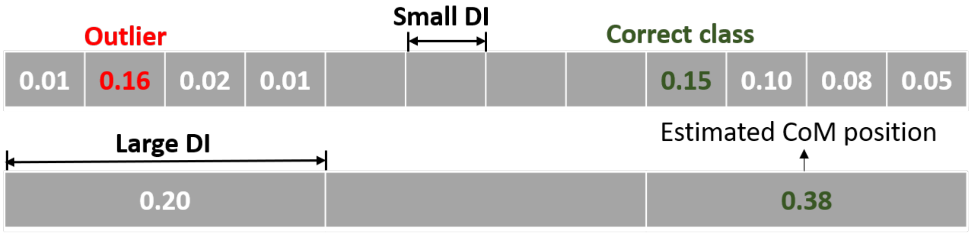


Figure 4.7: Trade-off on the selection of discretization interval (DI).

To solve the above problems, we propose a coarse-to-fine approach to avoid outliers and improve the accuracy in CoM estimation. First, we train several CNN models with different DIs in descending order ($DI_1 > DI_2 > \cdots$) and DI_k should be a multiple of DI_{k-1} (i.e., $DI_k = m_k \times DI_{k-1}$ where m_k is an integer). As larger DI ensures higher accuracy in the classification problem, we first use the model with the largest DI to decide the coarse range of the CoM position. Then, instead of directly using the center of the interval as the output, we use the model with smaller DI to obtain finer estimation of the CoM position. Figure 4.8 shows an example of three models ($DI_1 = 2DI_2 = 6DI_3$). We start with Model 1 and select the class with the highest probability (shown in green box). Then we use Model 2 and select between the two sub-classes that lie in the selected range resulting from Model 1. Similarly, we use Model 3 and select between the three sub-sub-classes that lie in the selected range resulting from Model 2. In this way, the outliers that may exist in the fine model (with small DI) are excluded in the coarse model (with large DI) and the precision of CoM estimation is improved in each step as the DI goes smaller. For the last model (with the smallest DI), we will output the final CoP position as the center of the selected small interval. Although the inference time will increase by using multiple models in the proposed coarse-to-fine approach, the inference time of each model is negligible (about 13 *ms*, see Table 4.2) by using the proposed colored skeleton image (proposed in Section 4.3.2 and validated in Section 4.4.3). Therefore, the total inference time on multiple models is also very small (< 40 *ms*, see Table 4.2) by using the proposed coarse-to-fine approach.

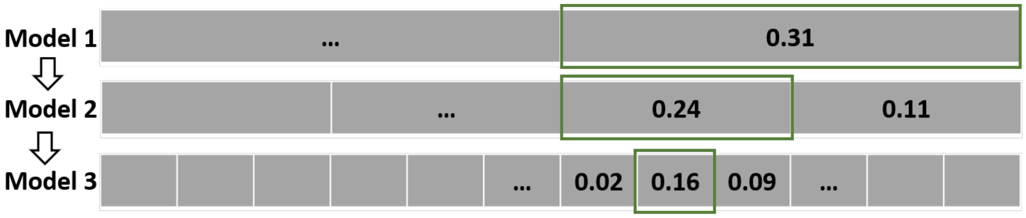


Figure 4.8: An example of the proposed coarse-to-fine approach. The green box represents the selected class in each model.

4.3.3 The Proposed Balance Evaluation System

Based on the CoM estimation model, we further propose a balance evaluation system to provide quantitative balance evaluation using the GI task. The subject's depth images and CoP positions are captured by the Kinect camera and the WBB (see Section 4.3.1). The subject's CoM positions are estimated from the depth images using the proposed CoM estimation model. As GI is a dynamic posture, the subject's CoP position is not equivalent to the CoM position. As proposed in [85], the maximum distance between the subject's CoP and CoM position during GI is correlated with the subject's dynamic balance control. Therefore, we calculate the CoP-CoM distance during the GI task. An example of the CoP-CoM trajectory and the CoPCoM distance vs. time in the x/y direction and the 2D distance (i.e., distance in the xy plane) during GI is shown in Figure 4.9. The right foot is the stepping foot. The subject's motion during GI can be divided into three states $S_1 \sim S_3$. In S_1, the CoP of the subject shifts towards the stepping foot and the CoM remains at the original position, therefore the CoP-CoM distance increases. In S_2, the subject's CoP shifts back towards the standing limb, as the stepping limb advances. During this time, the CoP-CoM distance first decreases and then increases. In S_3, the subject's CoP and CoM both move forward and the CoP-CoM distance continues to increase. From Figure 4.9 we can see that the maximum CoP-CoM distance occurs at the end of S_3. To build the balance evaluation model, we propose to extract the following features from the subject's CoP-CoM trajectory during the GI task.

- The maximum 2D CoP-CoM distance.
- The range of motion of the subject's CoM, in the x- and y-direction separately.

In our data collection process, each subject was required to perform three repetitions of GI on each leg. The motion of each subject (including all the six repetitions) constitutes a data sample. Therefore, there are $3 \times 6 = 18$ features in the input for each sample. Similar to the CoM estimation model, we propose a data augmentation approach for the balance evaluation model

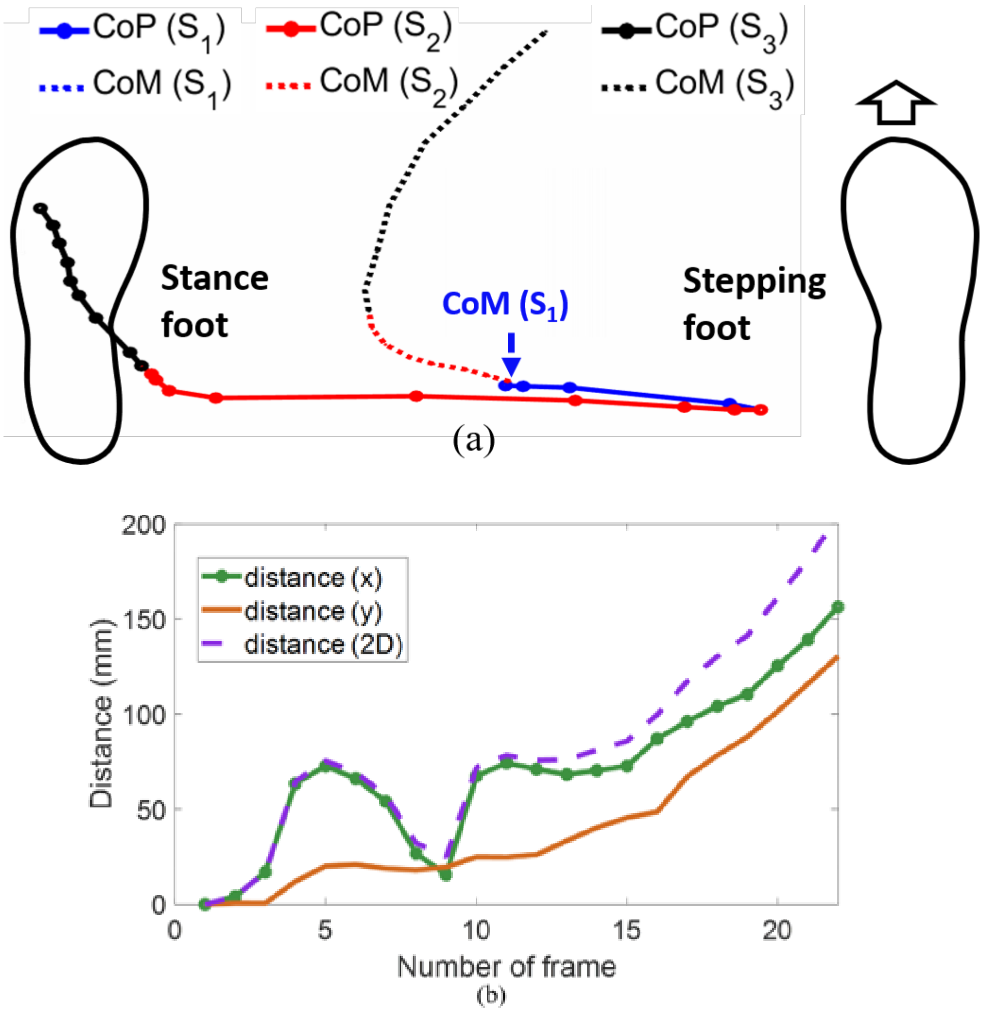


Figure 4.9: (a) The CoP-CoM trajectory during the GI task in three states. S_1: CoP shifts towards the stepping foot and CoM remains at the original position; S_2: CoP shifts back towards the standing limb; S_3: both CoP and CoM move forward. (b) The CoP-CoM distance vs. frame number during the GI task.

to create more training samples and avoid over-fitting. For the three repetitions that a subject performs on the left leg (e.g., L1, L2, L3), the order of the repetitions does not affect the overall performance of the subject and the PT's evaluation. Therefore, the output of this sample should remain unchanged if the order of the three repetitions on the left leg is changed (e.g., L3, L1, L2). Based on the above insight, we propose the following data augmentation approach. For the three repetitions on the left leg, there are $3! = 6$ types of permutations. Similarly, there are six types of permutations for the three repetitions on the right leg. Therefore, we can generate $6 \times 6 = 36$

samples from each original sample by changing the order of the repetitions. We propose to train a Random Forest (RF) classifier [95] to estimate a balance level from the input features. During the data collection, the subject's balance ability was tested clinically by the PT with the mini-BESTest and used as the ground truth. The mini-BESTest scores were classified into four levels as follows.

- Level 4 (score 28): no balance problem, no fall risk
- Level 3 (score 18 $\sim$ 27): mild balance problems, no fall risk.
- Level 2 (score 11 $\sim$ 17): medium balance problems, medium fall risk.
- Level 1 (score 0 $\sim$ 10): severe balance problems, high fall risk.

The balance level calculated from the PT score was used as the ground truth to train the balance evaluation model. The RF classifier takes all the 18 features as input and provides an estimate of the balance level as the output. Based on the study of Leddy et al. [98], patients with PD who get a score lower than 63% of the total score (i.e., $28 \times 63\% = 17.6$) on the mini-BESTest have fall risk. Therefore, Level 1 and 2 in our proposed balance evaluation system indicates fall risk. By using the proposed balance evaluation system, the patient is able to monitor his/her balance level and fall risk using a portable depth camera and WBB at home or any other place, which enables on-demand balance evaluation.

4.4 Results

In this section, we will first present our data collection process, then introduce the implementation details, finally evaluate the performance of the proposed CoM estimation and balance evaluation system.

4.4.1 Data Collection

This study was approved by the Institutional Review Board at University of California San Diego (protocol #181413X). 41 subjects (age 23 $\sim$ 81, 26 males, 15 females) participated in this study, including 21 healthy subjects and 20 patients with PD. To validate that our proposed model is able to learn the body parameters of the subject, we have recruited subjects of different body types (height 155 $\sim$ 190 *cm*, weight 44 $\sim$ 96 *kg*). All subjects signed the informed consent form. There were two stages in our data collection process. In the first stage, we collected data to train and test the proposed CoM estimation model. Each subject stood on the WBBs (shown in Figure 4) and performed the following static postures on four body parts.

Trunk: keep it upright, or lean to left/right/front/back with different angles.

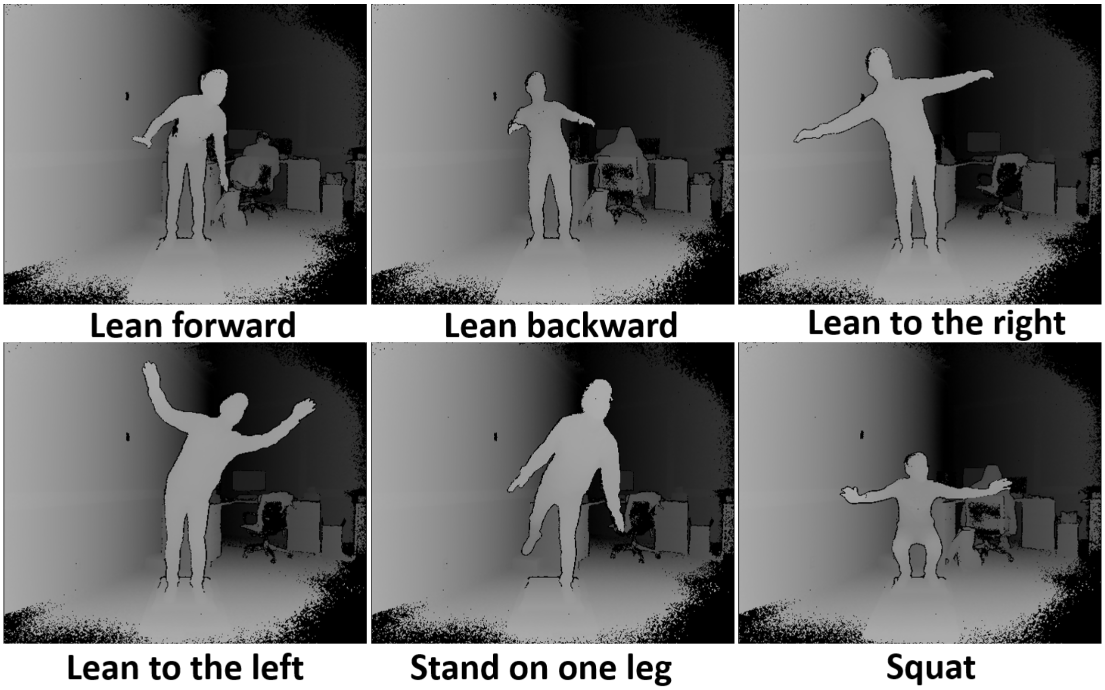
Legs: squat with different angles, stand on one leg.

Arms: different positions of the left and right arm.

Feet: different positions of the left and right foot.

Figure 10 shows some examples of the postures we have collected in our data collection.

The two WBBs recorded the CoP position, which was equivalent to the horizontal CoM position. We also used a Kinect sensor to capture the depth images of the subject. The WBB and the Kinect sensor were synchronized and the framerate was 30 frames per second. In the second stage, we collected data during the GI task for the balance evaluation system. Each subject stood on the WBB #1, made a step forward on the WBB #2 according to his/her natural walking, and steadily stepped off the board. Each subject performed three repetitions on the left and right leg separately. The CoP positions and depth images were also recorded by the WBB and the Kinect camera. The subject's dynamic balance was tested using the mini-BESTest by the PT as the ground truth.



Lean forward

Lean backward

Lean to the right

Lean to the left

Stand on one leg

Squat

Figure 4.10: Examples of static postures collected in our experiments.

4.4.2 Implementation Details

For the CoM estimation model, we used [-40, 40] (pixels), [-0.2, 0.2] (depth value), and [-15, 15] (pixels), for the random shift in the *x*-, *y*- (depth), and *z*-direction in the data augmentation. In the heatmap of the ground-truth CoM, we used Gaussian distribution with standard deviation of 3 and 2, in the *x*- and *y*-direction. There are five Conv units in the CNN-based model. In each Conv unit, 8, 16, 32, 64, 128 channels were used for the Conv layer respectively. The number of channels was selected empirically. The BN momentum was set to 0.9. When training the model, we used an Adam optimizer [96] to minimize the cross-entropy loss. The batch size was 64 and the learning rate was $5e-4$. For the proposed coarse-to-fine approach, we trained three models using $DI_1 = 8mm$, $DI_2 = 4mm$, and $DI_3 = 2mm$. For the balance evaluation model, we trained a RF classifier with 300 trees in the forest. The input of the classifier is 18-dimensional and the output is four categories. We used Gini impurity to measure the quality of a split when constructing the trees.

4.4.3 CoM Estimation Results

To validate the proposed CoM estimation model, we calculate the estimation error as the distance between the ground-truth CoM position and the estimated position (i.e., the center of the output class). Firstly we validate the performance of the model on existing subjects. We randomly split all the samples into three parts: a training set (including 64% of the samples), a validation set (including 16% of the samples), and a test set (including the rest 20% of the samples). Secondly we validate the performance of the model on a new subject. The samples of 40 subjects are used for training and validation and the samples from the 41st subject are used for testing. This process is repeated for 10 times and the average results are presented. We compare the results of the following methods: the CNN-based model proposed in our preliminary work [86], the CNN + coarse-to-fine approach proposed in this chapter, and two state-of-the-art methods: the SESC method [87] and the voxel reconstruction method [79]. Table 4.1 presents the estimation error and the requirements of each method. Fig 4.11 shows the error distribution of the proposed CNN + coarse-to-fine approach when testing on existing subjects and a new subject. The *x*-axis shows the CoM estimation error (*mm*) and the *y*-axis shows the normalized probability.

Table 4.1: CoM estimation error and requirement of each method.

Method		Error (mm)		Requirements
		x	*y*	
SESC [87]		17	23	Motion capture sensor. Identification needed for each new subject.
Voxel reconstruction [79]		8 ∼ 15		Five cameras. Camera calibration and synchronization.
CNN-based (our previous work) [86]	Existing subjects	6.0	9.3	Single depth camera. No calibration or identification needed.
	New subject	8.9	17.2	
CNN + coarse-to-fine (proposed)	Existing subjects	5.2	8.4	
	New subject	7.8	15.7	

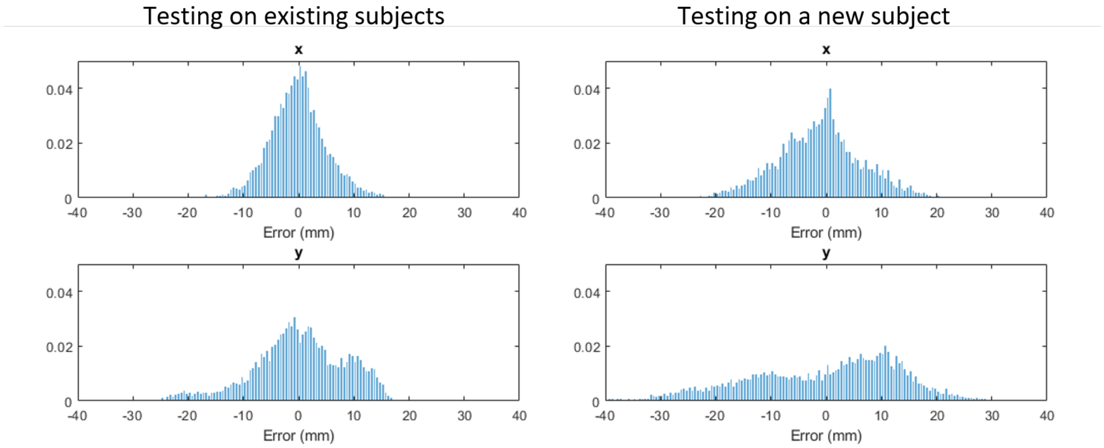


Figure 4.11: Error distribution of the proposed CNN + coarse-to-fine approach.

When testing on existing subjects, the proposed CNN-based method (proposed in [86] and in this chapter) achieves the lowest estimation error. When testing on a new subject, the estimation error achieved by our methods increases a little bit, but still outperforms the SESC method in both *x*- and *y*- directions. In additional, the identification phase required by the SESC method is not convenient for home and clinical use. For example, an existing subject may need to go through the identification phase again if he/she gains or loses weight. In comparison, the proposed CNN-based approach is able to learn the subject's body parameters from the depth image without any identification process. Compared with the voxel reconstruction method [79], our proposed approach achieves comparable accuracy results, but requires only a single depth camera and avoids the complicated calibration and synchronization among multiple cameras. Therefore, it is more convenient for home and clinical use. Comparing the estimation error in the *x*- and *y*-direction, we can see that the error in the *y*-direction (depth direction) is higher, which is due to the fact that the back side of the body cannot be captured by the single depth camera.

Moreover, the coarse-to-fine approach proposed in this chapter further reduces the estimation error by about 10%, compared with the preliminary model in [86]. Besides, TABLE 4.2 shows the comparison of the total training time (i.e., the total time to update the parameters in one epoch) and the average inference time (i.e., the average time on each sample) by using the proposed colored skeleton image (discussed in Section 4.3.2) and the joint heatmaps proposed in [86]. For the colored skeleton image approach, we show the training and inference times using single model and multiple models in the coarse-to-fine approach. The running time is tested on an Intel Xeon E5-1650 CPU and an NVIDIA GeForce GTX 1080 Ti GPU. We can see that the training and inference times of single model are significantly reduced by using the proposed colored skeleton image in the input of the CNN model. Although the proposed coarse-to-fine approach increases the training and inference times by using multiple models, it can still achieve much less training time and comparable inference time compared with the preliminary model proposed in [86], while significantly reducing the estimation error (see Table 4.1). Therefore, it can be concluded that the CoM estimation model proposed in this chapter improves our preliminary model proposed in [86] by significantly reducing the estimation error, as well as the training and inference times.

Table 4.2: Comparison of the training and inference times.

Feature		**Total training time (s)**	**Average inference time (ms)**
Joint heatmaps [86]		2252.1	38.0
Colored skeleton image (proposed)	Single model	322.7	12.6
	Coarse-to-fine	963.5	38.6

4.4.4 Balance Evaluation Results

To show the performance of the proposed balance evaluation system, we first provide more details on the collected data during the GI task. Table 4.3 shows the average value of each input feature (discussed in Section 4.3.3) for each balance level. We can see that subjects in lower balance level (i.e., worse balance) have smaller CoP-CoM distance. Similarly, subjects with worse balance also show smaller range of motion in their CoM position in the y-direction (i.e. the anterior-posterior direction), which indicates that subjects with worse balance have smaller step length and smaller body movement during the GI task. For the range of motion in the *x*-direction (i.e., the medio-lateral direction), subjects in level 4 (who got full score 28 in the mini-BESTest) have higher range of motion. However, there is no significant trend for the other three levels.

Table 4.3: Average feature values for each balance level.

Feature		**Balance level**			
		1	2	3	4
Maximum CoP-CoM distance (mm)		137.3	177.3	217.8	264.2
Range of motion of the CoM position (mm)	*x*	22.1	18.4	15.1	50.0
	y	86.3	115.8	178.7	74.1

To validate the proposed RF-based balance evaluation model, we conduct experiments using 10-fold cross validation, with 90% of the collected samples used for training and 10% for testing. The proposed data augmentation approach is applied to the training samples. We calculate the sensitivity (i.e., the proportion of actual positive samples that are correctly classified) and specificity (i.e., the proportion of actual negative samples that are correctly classified) for each level and report the results in Table 4.4. We also show the results on the two categories: with fall risk (levels 1 and 2) and without fall risk (i.e., levels 3 and 4). We can see that the proposed RF-based model can achieve high sensitivity and specificity for the four levels (> 80%) and the two categories (> 90%). Besides, all the classification error is only one level (i.e., no sample is misclassified as a level higher or lower than the ground-truth level by two levels or more). Therefore, it can be concluded that the proposed balance evaluation system is able to provide accurate and quantitative balance assessments like a human PT. The high accuracy also demonstrates that the proposed CoM estimation model works for dynamic postures. By using the proposed balance evaluation system, the patient can measure his/her balance level using a simple GI task at home or in the clinic. The quantitative balance level can help the patient (and his/her PT) evaluate progress in physical therapy training, select the proper training programs, and predict the fall risk.

Table 4.4: Sensitivity and specificity using the proposed balance evaluation model.

	Balance level				**Fall risk**	
	1	2	3	4	Yes	No
Sensitivity	87.5%	82.5%	82.0%	85.0%	93.8%	95.4%
Specificity	98.8%	93.5%	89.4%	97.7%	95.4%	93.8%

4.5 Conclusion

In this chapter, we propose a balance evaluation system using camera and WBB sensors to enable on-demand balance evaluation for home and clinic-based physical therapy. To develop this system, we first propose a CoM estimation model to estimate the CoM position of the human body from a depth image. Experimental results on the CoM estimation model demonstrate its superiority over other CoM estimation techniques, including high accuracy and the ease-of-use. Based on the CoM estimation model, we further propose the balance evaluation system to estimate a quantitative balance level from the subject's performance during a GI task. Experimental results show that the proposed model can accurately estimate a balance level that is consistent with the human PT's evaluation in traditional balance tests. By using portable and inexpensive sensors, the proposed balance evaluation system enables on-demand balance evaluation for home and clinical use and has the potential of significantly reducing clinic visit requirements and reducing cost for both the patients and care providers.

Chapter 4, in part, is from the material as it appears in proceedings of IEEE International Conference on Healthcare Informatics 2019, Wenchuan Wei; Sujit Dey. and in IEEE Access 2020. Wenchuan Wei; Carter McElroy; Sujit Dey. The dissertation author was the primary investigator and author of the papers.

Chapter 5

Conclusion and Future Work

In this dissertation, we propose a virtual PT model using multiple sensors and AI to enable on-demand training, monitoring, task recommendation, and balance evaluation for physical therapy. We have collected real patient data from offline sessions and trained a virtual PT model. The patient can use a mobile device to access the virtual PT model remotely. Avatar-based instructions and guidance are rendered on the cloud and sent to the patient's device in real time. During live home sessions, multiple sensors are used to track the patient's movements and performance. We have proposed algorithms to evaluate the patient's performance, identify the movement error, and provide task recommendations. To track the patient's progress and validate the effectiveness of the training programs, we have also proposed a balance evaluation model, which can quantify the patient's dynamic balance during a simple GI task. All the proposed algorithms are trained from real patient data collected by human PTs. Experimental results have shown the accuracy of the proposed system and its superiority over the other techniques. By using inexpensive sensors and AI, the proposed virtual PT system has the advantages of providing accurate, on-demand and personalized care.

In the future, we would like to extend our research in the following directions. Firstly, the skeleton tracking results of the Kinect sensor are sometimes inaccurate and unstable, especially when the user is performing some complicated movements or using walkers and wheelchairs. Besides, the Kinect sensor requires that the user should stand/sit in front of the camera (i.e., the front view) with a distance of 0.5 $\sim$ 4.5 meters, which limits its application. Therefore, we would like to improve the tracking accuracy and enable more views by using multiple cameras or incorporating other sensors.

Secondly, the ground truth we used when training the models were manually labeled by our PT collaborator in this project. However, the ground truth may be inaccurate due to the PT's subjective bias. In the future, it will be helpful to invite multiple PTs to label the same patient data independently and combine their annotations as the ground truth.

Thirdly, we would like to improve the proposed CoM estimation and balance evaluation model. We would like to test the accuracy of the WBB by comparing it with a laboratory-grade force plate in our data collection. We also plan to improve the current balance evaluation system to provide more detailed balance assessments (e.g., continuous balance scores) instead of the four levels. Besides, the GI task discussed in this study may be limited for balance evaluation. We plan to explore more training exercises in physical therapy to achieve more comprehensive balance evaluation for patients with balance problems.

Last but not least, we would like to explore more about the design of guidance. Currently the proposed visual and textual guidance are proved useful for the user to improve performance, and the combination of visual and textual guidance is the most helpful (discussed in Chapter 2). However, many other issues need to be considered to improve the effectiveness of guidance, e.g., are there other types of guidance which may be more effective for certain types of patients, what is the proper frequency to provide guidance, and how much guidance might be appropriate as opposed to being overwhelming for the user. All of these issues need to be considered and explored in our future work.

Bibliography

[1] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, “An IoT-aware architecture for smart healthcare systems,” *IEEE Internet of Things Journal* 2.6 (2015): 515-526.

[2] Z. Yang, Q. Zhou, L. Lei, K. Zheng, and W. Xiang, “An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare,” *Journal of medical systems* 40.12 (2016): 286.

[3] K. Aziz, S. Tarapiah, S. H. Ismail, and S. Atalla, “Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies,” *Big Data and Smart City* (ICBDSC’16), Muscat, March, 2016.

[4] Z. Ali, G. Muhammad, and M. F. Alhamid, “An Automatic Health Monitoring System for Patients Suffering From Voice Complications in Smart Cities,” *IEEE Access* 5 (2017): 3900-3908.

[5] P. Choden, T. Seesaard, T. Eamsa-Ard, C. Sriphrapradang, and T. Kerdcharoen, “Volatile urine biomarkers detection in type II diabetes towards use as smart healthcare application,” *Knowledge and Smart Technology* (KST’17), Chonburi, February, 2017.

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

[7] D. Jack, R. Boian, A. S. Merians, M. Tremaine, G. C. Burdea, S. V. Adamovich, M. Recce, H. Poizner, “Virtual reality-enhanced stroke rehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9.3(2001): 308-318.

[8] A. Mirelman, B. L. Patritti, P. Bonato, and J. E. Deutsch, “Effects ofvirtual reality training on gait biomechanics of individuals post-stroke,” *Gait & posture*, 31.4 (2010): 433-437.

[9] Mobile App Download and Usage Statistics. [Online]. Available: https://buildfire.com/app-statistics/.

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

[11] Y. Lu, Y. Liu, and S. Dey, “Cloud mobile 3D display gaming user experience modeling and optimization by asymmetric graphics rendering,” *IEEE Journal of Selected Topics in Signal Processing* 9.3 (2015): 517-532.

[12] Healthcare Weekly. [Online]. Available: https://healthcareweekly.com/artificial-intelligence-healthcare-market/.

[13] Unity. [Online]. Available: https://unity3d.com/

[14] W.Wei, Y.Lu, C.Printz, and, S.Dey, “Motion Data Alignment and Real-Time Guidance in Cloud-Based Virtual Training System,” *in Proc. of Wireless Health* (WH’15), Bethesda, Oct. 2015.

[15] S. Ananthanarayan, M. Sheh, A. Chien, H. Profita, and K. Siek, “Pt Viz: towards a wearable device for visualizing knee rehabilitation exercises," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI’13), Paris, April, 2013.

[16] C. Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, and A. A. Rizzo, “Towards pervasive physical rehabilitation using Microsoft Kinect,” *Pervasive Computing Technologies for Healthcare* (PervasiveHealth’12), San Diego, May, 2012.

[17] B. Lange, C. Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas, “Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor,” *Engineering in Medicine and Biology Society* (EMBC’11), Boston, September, 2011.

[18] Y. J. Chang, S. F. Chen, and J. D. Huang, “A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities,” *Research in developmental disabilities* 32.6 (2011): 2566-2570.

[19] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice, “YouMove: enhancing movement training with an augmented reality mirror,” *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST’13), St Andrews, October, 2013.

[20] 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. of the 19th ACM international conference on Multimedia* (MM’11), Scottsdale, November, 2011.

[21] A. Yurtman, and B. Barshan, “Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units,” *Information Sciences and Systems* (SIU’14), Trabzon, April, 2014.

[22] O. Bau, and W. E. Mackay, “OctoPocus: a dynamic guide for learning gesture-based command sets,” *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST’08), Monterey, October, 2008.

[23] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor, “ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures,” *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS’09), Banff, November, 2009.

[24] R. Sodhi, H. Benko, and A. Wilson, “LightGuide: projected visualizations for hand movement guidance,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI’12), Austin, May, 2012.

[25] J. Doyle, C. Bailey, B. Dromey, and C. N. Scanaill, “BASE-An interactive technology solution to deliver balance and strength exercises to older adults,” *2010 4th International Conference on Pervasive Computing Technologies for Healthcare* (PervasiveHealth’10), Munich, March, 2010.

[26] R.Tang, X.D.Yang, S.Bateman, J.Jorge, and A.Tang, “Physio@ Home: Exploring visual guidance and feedback techniques for physiotherapy exercises,” *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI’15), Seoul, April, 2015.

[27] The Microsoft documentation for Kinect 2.0. [Online]. Available: https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx

[28] S. Dey, Y. Liu, S. Wang, and Y. Lu, “Addressing response time of cloud- based mobile applications,” *Proceedings of the first international workshop on Mobile cloud computing & networking*, ACM, 2013.

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

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

[31] K. Kahol, P. Tripathi, and S. Panchanathan, “Automated gesture segmentation from dance sequences,” Proceedings of the Sixth IEEE International conference on Automatic Face and Gesture Recognition (FGR’04), Seoul, May, 2004.

[32] D. Kim, J. Song, and D. Kim, “Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs,” *Pattern recognition*, 40.11 (2007): 3012-3026.

[33] G. A. Seber, and A. J. Lee, “Linear regression analysis,” Vol. 936. John Wiley & Sons, 2012.

[34] Linktropy. [Online]. Available: http://www.apposite-tech.com/products/

[35] K. L. Gwet, "Intrarater reliability," *Wiley encyclopedia of clinical trials*, 2008.

[36] Amazon Web Services. [Online]. Available: https://aws.amazon.com

[37] Parkinson’s disease statistics by Parkinson’s Foundation. [Online]. Available: http://parkinson.org/Understanding-Parkinsons/Causes-and- Statistics/Statistics

[38] L. A. King, J. Wilhelm, Y. Chen, R. Blehm, J. Nutt, Z. Chen, A. Serdar, and F. B. Horak, “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.

[39] M. D. Hssayeni, J. L. Adams, and B. Ghoraani, “Deep Learning for Medication Assessment of Individuals with Parkinson’s Disease Using Wearable Sensors,” *In Proc. of 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC’18), Honolulu, HI, USA, July 2018.

[40] E. Stack, V. Agarwal, R. King, M. Burnett, F. Tahavori, B. Janko, W. Harwin, A. Ashburn, and D. Kunkel, “Identifying balance impairments in people with Parkinson’s disease using video and wearable sensors,” *Gait & posture* 62 (2018): 321-326.

[41] B. Galna, D. Jackson, G. Schofield, R. McNaney, M. Webster, G. Barry, D. Mhiripiri, M. Balaam, P. Olivier, and L. Rochester, “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.

[42] J. E. Pompeu, L. A. Arduini, A. R. Botelho, M. B. F. Fonseca, S. A. A. Pompeu, C. Torriani-Pasin, and J. E. Deutsch, “Feasibility, safety and outcomes of playing Kinect Adventures!TM for people with Parkinson’s disease: a pilot study,” *Physiotherapy* 100.2 (2014): 162-168.

[43] Z. Wang, J. Liao, Q. Cao, H. Qi and Z. Wang, "Friendbook: a semantic-based friend recommendation system for social networks,” *IEEE transactions on mobile computing* 14.3 (2015): 538-551.

[44] M. Yan, J. Sang, and C. Xu, “Unified youtube video recommendation via cross-network collaboration,” *in Proc. of the 5th ACM on International Conference on Multimedia Retrieval* (ICMR’15), Shanghai, China, Jun. 2015.

[45] W. Wei, C. McElroy, and S. Dey, “Human Action Understanding and Movement Error Identification for the Treatment of Patients with Parkinson’s Disease,” *in Proc. of IEEE International Conference on Healthcare Informatics* (ICHI’18), New York City, USA, Jun. 2018.

[46] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester, “Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson’s disease,” *Gait & posture* 39.4 (2014): 1062-1068.

[47] 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’13), Hong Kong, China, Jul. 2013.

[48] 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’12), Providence, RI, USA, Jun. 2012.

[49] 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’12), Saint Paul, MN, USA, Jun. 2012.

[50] 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’14), Columbus, OH, USA, Jun. 2014.

[51] 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 Recognitio*n (CVPR’15), Boston, MA, USA, Jun. 2015.

[52] 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.

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

[54] 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.

[55] 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.

[56] 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.

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

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

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

[60] L. Breiman, “Random forests,” *Machine learning* 45.1 (2001): 5-32.

[61] C. X. Ling, and C. Li, “Data mining for direct marketing: Problems and solutions,” *KDD*. Vol. 98. 1998.

[62] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of artificial intelligence research* 16 (2002): 321-357.

[63] Z. Yu, C. Wang, J. Bu, X. Wang, Y. Wu, and C. Chen, “Friend recommendation with content spread enhancement in social networks,” *Information Sciences* 309 (2015): 102-118.

[64] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, “Content-based video recommendation system based on stylistic visual features,” *Journal on Data Semantics* 5.2 (2016): 99-113.

[65] F. Provost, and T. Fawcett, “Robust classification for imprecise environments,” *Machine learning* 42.3 (2001): 203-2.

[66] M. Kubat, and S. Matwin, ”Addressing the curse of imbalanced training sets: one-sided selection,” *Icml.* Vol. 97. 1997.

[67] C. X. Ling, and C. Li, “Data mining for direct marketing: Problems and solutions,” *Kdd*. Vol. 98. 1998.

[68] M. K. Mak, and M. M. Auyeung, “The mini-BESTest can predict parkinsonian recurrent fallers: a 6-month prospective study,” *Journal of rehabilitation medicine* 45.6 (2013): 565-571.

[69] K. O. Berg, S. L. Wood-Dauphinee, J. I. Williams, and B. Maki, “Measuring balance in the elderly: validation of an instrument,” *Canadian journal of public health= Revue canadienne de sante publique* 83 (1992): S7-11.

[70] F. Franchignoni, F. Horak, M. Godi, A. Nardone, and A. Giordano, “Using psychometric techniques to improve the Balance Evaluation Systems Test: the mini-BESTest,” *Journal of rehabilitation medicine* 42.4 (2010): 323331.

[71] A. K. Mishra, M. Skubic, B. W. Willis, T. Guess, S. S. Razu, C. Abbott, and A. D. Gray, “Examining methods to estimate static body sway from the Kinect V2. 0 skeletal data: implications for clinical rehabilitation,” *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2017)*, Barcelona, Spain, May 2017.

[72] M. W. Kennedy, and J. P. Schmiedeler, C. R. Crowell, M. Villano, A. D. Striegel, and J. Kuitse, “Enhanced feedback in balance rehabilitation using the Nintendo Wii Balance Board,” *Proceedings of the IEEE International Conference on e-health networking, applications and services (Healthcom 2011)*, Columbia, USA, Jun. 2011.

[73] H. L. Bartlett, L. H. Ting, and J. T. Bingham, “Accuracy of force and center of pressure measures of the Wii Balance Board,” *Gait & posture* 39.1 (2014): 224-228.

[74] W. Young, S. Ferguson, S. Brault, and C. Craig, “Assessing and training standing balance in older adults: a novel approach using the ‘Nintendo Wii’Balance Board,” *Gait & posture* 33.2 (2011): 303-305.

[75] J. D. Holmes, M. E. Jenkins, A. M. Johnson, M. A. Hunt, and R. A. Clark, “Validity of the Nintendo Wii® balance board for the assessment of standing balance in Parkinson’s disease,” *Clinical Rehabilitation* 27.4 (2013): 361-366.

[76] D. A. Winter, *Biomechanics and motor control of human movement.* John Wiley & Sons, 2009.

[77] S. C. Chen, H. J. Hsieh, T. W. Lu, and C. H. Tseng, “A method for estimating subject-specific body segment inertial parameters in human movement analysis,” *Gait & posture* 33.4 (2011): 695-700.

[78] S. Cotton, A. P. Murray, P. Fraisse, “Estimation of the center of mass: from humanoid robots to human beings,” *IEEE/ASME Transactions on Mechatronics* 14.6 (2009): 707-712.

[79] T. Kaichi, S. Mori, H. Saito, K. Takahashi, D. Mikami, M. Isogawa, and H. Kimata, “Estimation of Center of Mass for Sports Scene Using Weighted Visual Hull,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2018)*, Salt Lake City, UT, USA, Jun. 2018.

[80] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, “Recurrent neural networks for emotion recognition in video,” *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICML 2015)*, Seattle, Washington, USA, Nov. 2015.

[81] R. Alazrai, Y. Mowafi, and E. Hamad, “A fall prediction methodology for elderly based on a depth camera,” *Proceedings of the IEEE Conference on Engineering in Medicine and Biology Society (EMBC 2015)*, Milan Italy, Nov. 2015.

[82] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, “Efficient human pose estimation from single depth images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (2013): 2821-2840.

[83] 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.

[84] W. Wei, C. McElroy, S. Dey, "Towards On-Demand Virtual Physical Therapist: Machine Learning-Based Patient Action Understanding, Assessment and Task Recommendation," *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, vol. 27, no. 9, pp. 1824-1835, Sept. 2019.

[85] C. J. Hass, D. E. Waddell, R. P. Fleming, J. L. Juncos, and R. J. Gregor, “Gait initiation and dynamic balance control in Parkinson’s disease,” *Archives of physical medicine and rehabilitation* 86.11 (2005): 2172-2176.

[86] W. Wei, and S. Dey, “Center of Mass Estimation for Balance Evaluation Using Convolutional Neural Networks,” *Proceedings of the Seventh IEEE International Conference on Healthcare Informatics (ICHI 2019)*, Xi’an, China, Jun. 2019.

[87] A. González, M. Hayashibe, V. Bonnet, and P. Fraisse, “Whole body center of mass estimation with portable sensors: Using the statically equivalent serial chain and a Kinect,” *Sensors* 14.9 (2014): 16955-16971.

[88] J. Swanenburg, E. D. de Bruin, K. Favero, D. Uebelhart, and T. Mulder, “The reliability of postural balance measures in single and dual tasking in elderly fallers and non-fallers,” *BMC musculoskeletal disorders* 9.1 (2008): 162.

[89] Kinect. [Online]. Available: https://www.xbox.com/en-US/kinect, Accessed on: Aug. 5, 2019.

[90] A. Huurnink, D. P. Fransz, I. Kingma, and J. H. van Dieën, “Comparison of a laboratory grade force platform with a Nintendo Wii Balance Board on measurement of postural control in single-leg stance balance tasks,” *Journal of biomechanics* 46.7 (2013): 1392-1395.

[91] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” *Advances in neural information processing systems (NIPS 2014)*, Montreal, Canada, Dec. 2014.

[92] Y. LeCun, and Y. Bengio, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks* 3361.10 (1995): 1995.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems (NIPS 2012)*, Lake Tahoe, USA, Dec. 2012.

[94] S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167* (2015).

[95] L. Breiman, “Random forests,” *Machine learning* 45.1 (2001): 5-32.

[96] D. P. Kingma, and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*(2014).

[97] S. E. Halliday, D. A. Winter, J. S. Frank, A. E. Patla, and F. Prince, “The initiation of gait in young, elderly, and Parkinson’s disease subjects,” *Gait & posture* 8.1 (1998): 8-14.

[98] A. L. Leddy, B. E. Crowner, and G. M. Earhart, “Utility of the Mini-BESTest, BESTest, and BESTest sections for balance assessments in individuals with Parkinson disease,” *Journal of neurologic physical therapy: JNPT* 35.2 (2011): 90.

[99] Y. L. Hsu, P. C. Chung, W. H. Wang, M. C. Pai, C. Y. Wang, C. W. Lin, H. L. Wu, and J. S. Wang, “Gait and balance analysis for patients with Alzheimer’s disease using an inertial-sensor-based wearable instrument,” *IEEE journal of biomedical and health informatics* 18.6 (2014): 1822-1830.

[100] P. Esser, H. Dawes, J. Collett, and K. Howells, “IMU: inertial sensing of vertical CoM movement,” *Journal of biomechanics* 42.10 (2009): 1578-1581.

[101] B. Fasel, J. Spörri, P. Schütz, S. Lorenzetti, and K. Aminian, “An inertial sensor-based method for estimating the athlete’s relative joint center positions and center of mass kinematics in alpine ski racing,” *Frontiers in physiology* 8 (2017): 850.

[102] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Hawaii, USA, Jul. 2017.

[103] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *European conference on computer vision (ECCV 2016)*, Amsterdam, Netherlands, Oct. 2016.