

# RECOGNITION OF OCCUPATIONAL THERAPY EXERCISES FOR CEREBRAL PALSY

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE  
OF BILKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF  
MASTER OF SCIENCE  
IN  
COMPUTER ENGINEERING

By  
Mehmet Faruk Ongun  
September 2018

RECOGNITION OF OCCUPATIONAL THERAPY EXERCISES  
FOR CEREBRAL PALSY

By Mehmet Faruk Ogun

September 2018

We certify that we have read this thesis and that in our opinion it is fully adequate,  
in scope and in quality, as a thesis for the degree of Master of Science.

---

Uğur Güdükbay(Advisor)

---

Selim Aksoy

---

Ramazan Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

---

Ezhan Kardeşan  
Director of the Graduate School

## ABSTRACT

# RECOGNITION OF OCCUPATIONAL THERAPY EXERCISES FOR CEREBRAL PALSY

Mehmet Faruk Ongun

M.S. in Computer Engineering

Advisor: Uğur Güdükbay

September 2018

Depth camera-based virtual rehabilitation systems are gaining traction in occupational therapy for approaching patients with cerebral palsy. When developing such a system, a domain specific exercise recognition method is vital. In order to design a successful gesture recognition solution for this specific purpose, some obstacles needs to be overcome, namely; detection of gestures that are not related to the defined exercise set and recognition of incorrect exercises that are performed by the patients to compensate for their lack of ability. A combination of solutions, that are based on hidden Markov models, targeting aforementioned obstacles are proposed and elaborated on. The proposed solution works for upper extremity functional exercises and critical compensation mistakes together with restrictions for classifying these mistakes are determined with the help of occupational therapists. Afterwards, we first aim to eliminate the undefined gestures by designing two models that produce adaptive threshold values. Then, we utilize specific negative models based on an approach named feature thresholding and train them specifically for each exercise to distinguish the compensation mistakes. We conducted various tests using our method in a laboratory environment under the supervision of occupational therapists and presented the results of our proposed approach.

*Keywords:* gesture recognition, cerebral palsy, occupational therapy, hidden markov model, exercise recognition, depth camera, virtual rehabilitation.

## ÖZET

# SEREBRAL PALSİ HAŞTALIĞINA YÖNELİK ERGOTERAPİ EGZERSİZLERİNİN TANINMASI

Mehmet Faruk Ogun

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Danışmanı: Uğur Güdükbay

Eylül 2018

Serebral palsili hastaların rehabilitasyonuna yönelik olarak ergoterapistler tarafından derinlik kamerası tabanlı sanal rehabilitasyon uygulamalarının kullanımı ilgi çekmekte olan bir yaklaşımdır. Bu tip bir sistem geliştirirken, hedeflenen alana özel bir egzersiz tanıma yönteminin kullanılması oldukça önemlidir. Bu amaca yönelik, başarılı bir hareket tanıma çözümü geliştirmek için bazı problemlerin aşılması gerekmektedir. Bu problemler kısaca egzersiz kümesinde tanımlı olmayan hareketlerin tespit edilerek dikkate alınmaması ve hastalar tarafından fiziksel olarak yetersiz oldukları egzersizlerde bu eksiklerini telafi etme amacıyla yanlış bir şekilde tamamladıkları egzersizlerin hatalı olarak tanınabilmesidir. Saklı Markov model tabanlı olarak geliştirilen bir çözümler bütünü bahsedilen bu sorunlara çözüm olarak sunulmuştur. Geliştirilen çözüm üst ekstremite fonksiyonel egzersizleri ile çalışmaktadır. Çözümümüz, ilk olarak tanımlı olmayan hareketleri uyarlanmış eşik değeri üretebilmek üzere tasarlanan modeller aracılığıyla tespit ederek, elemektedir. Sonraki adımda ise ergoterapistler tarafından belirlenen telafi hataları ve bu hataların sınıflandırılabilmesi için gerekli kısıtlar kullanılarak öznel eşikleme adımı verdiğimiz yöntem üzerinde çalışan özel negatif modeller aracılığıyla bu telafi hatalarının tespiti sağlanır. Geliştirilen yöntemeye yönelik çeşitli testler uzman ergoterapistlerin gözetimi altında ve laboratuvar ortamında tamamlanmış ve elde edilen sonuçlar sunulmuştur.

*Anahtar sözcükler:* hareket tanıma, serebral palsi, ergoterapi, saklı markov model, egzersiz tanıma, derinlik kamerası, sanal rehabilitasyon.

## Acknowledgement

First and foremost, I would like to thank my supervisor, Prof. Dr. Uğur Gdkbay, for his guidance during my work. His support and dedication helped me to be motivated throughout my academic research.

I am deeply thankful to Assoc. Prof. Dr. Selim Aksoy and Asst. Prof. Dr. Ramazan Gkberk Cinbiş for kindly accepting to read and review this thesis.

I would also express my gratitude to my friends, especially Arif Usta and Hasan Balcı, for their invaluable help and patience.

I appreciate the support and understanding of my father, mother and sisters.

Finally, I would like to dedicate this thesis to my wife Zehra and my daughter, who gave me the needed tenacity to complete my work. Thank you for always being and bearing with me.

# Contents

- 1 Introduction** **1**
  - 1.1 Motivation and Scope . . . . . 1
  - 1.2 Contributions . . . . . 3
  - 1.3 Organization of the Thesis . . . . . 3
  
- 2 Background and Related Work** **4**
  - 2.1 Gesture Recognition . . . . . 4
    - 2.1.1 Hidden Markov Models . . . . . 6
  - 2.2 Occupational Therapy . . . . . 14
    - 2.2.1 Cerebral Palsy . . . . . 17
    - 2.2.2 Exercise Set . . . . . 17
  - 2.3 Depth Cameras . . . . . 21
    - 2.3.1 Kinect v1 and Kinect v2 . . . . . 22
    - 2.3.2 Kinect in Rehabilitation . . . . . 23

2.4	Definition of the Setup . . . . .	24
<b>3</b>	<b>Gesture Recognition Model</b>	<b>27</b>
3.1	Feature Selection . . . . .	29
3.1.1	Feature Count . . . . .	31
3.2	Hidden Markov Model Structure . . . . .	31
3.3	Scaling Problem . . . . .	33
3.4	Continuous Hidden Markov Model . . . . .	34
<b>4</b>	<b>Detection of Non-gesture Patterns</b>	<b>35</b>
4.1	Universal Negative Model . . . . .	36
4.2	Universal Positive Model . . . . .	37
4.3	Threshold Model . . . . .	38
4.4	Comparison . . . . .	40
<b>5</b>	<b>Improved Accuracy for Incorrect Exercises</b>	<b>42</b>
5.1	Feature Thresholding . . . . .	43
5.2	Negative Models . . . . .	45
5.2.1	Fault-specific Negative Model . . . . .	45
5.2.2	Gesture-specific Negative Model . . . . .	46
5.2.3	Comparison . . . . .	46

<i>CONTENTS</i>	viii
<b>6 Evaluation and Results</b>	<b>49</b>
<b>7 Conclusions and Future Research Directions</b>	<b>52</b>
<b>Bibliography</b>	<b>54</b>



# List of Figures

2.1	A model that corresponds to the tossing of two different coins. . .	7
2.2	Ergodic Hidden Markov Model . . . . .	12
2.3	Left-to-Right Hidden Markov Model . . . . .	12
2.4	Parallel Left-to-Right Hidden Markov Model . . . . .	12
2.5	The overview of the method proposed in [1]. . . . .	14
2.6	The framework of the recognition system proposed in [2]. . . . .	15
2.7	Shoulder flexion: starting position (1), arm is raised $180^\circ$ from the front while keeping the elbow angle as $180^\circ$ (2), and the end position (3). . . . .	19
2.8	Shoulder abduction: starting position (1), arm is raised $90^\circ$ from the side while keeping the elbow angle as $180^\circ$ (2), and the end position (3). . . . .	19
2.9	External rotation: starting position (1), arm is raised from the side while keeping the elbow and shoulder angles as $90^\circ$ (2), the arm is rotated using only the shoulder joint (3), the arm is rotated back (4), and end position (5). . . . .	19

2.10	Elbow flexion and extension: starting position (1), the arm is raised to the side while keeping the elbow straight and the shoulder angle as $90^\circ$ (2), the elbow is flexed to $60^\circ$ (3), the elbow is extended back (4), and the end position (5). . . . .	20
2.11	Combined PNF pattern: starting position (1), arm is raised diagonally while keeping the diagonal movement line straight (2), and back to the end position (3). . . . .	20
2.12	Kinect v1. . . . .	21
2.13	Kinect v2. . . . .	22
2.14	The comparison of the OptiTrack motion capture system and the coordinate data provided by Kinect [3]. . . . .	24
3.1	The framework of the proposed solution . . . . .	28
3.2	The graph depicting the feature count versus F1 score. . . . .	32
3.3	The proposed HMM structure. . . . .	33
4.1	The structure of the universal negative model. . . . .	37
4.2	A simplified structure of the threshold model [4]. . . . .	39
6.1	The success rate graph for (a) shoulder flexion, (b) shoulder abduction, (c) external rotation, (d) elbow flexion and extension, and (e) combined PNF pattern. . . . .	51

# List of Tables

2.1	The summary of notations. . . . .	7
2.2	The comparison of Kinect v1 and v2. . . . .	22
2.3	The validation set. . . . .	26
2.4	The training set. . . . .	26
4.1	The comparison of the models for shoulder flexion. . . . .	41
4.2	The comparison of the models for shoulder abduction. . . . .	41
4.3	The comparison of the models for external rotation. . . . .	41
4.4	The comparison of the models for elbow flexion and extension. . . . .	41
4.5	The comparison of the models for combined PNF pattern. . . . .	41
4.6	The overall comparison of the models. . . . .	41
5.1	The fault-specific negative model and the baseline solution for shoulder flexion. . . . .	47
5.2	The fault-specific negative model and the baseline solution for shoulder abduction. . . . .	47

5.3	The fault-specific negative model and the baseline solution for external rotation. . . . .	47
5.4	The fault-specific negative model and the baseline solution for elbow flexion and extension. . . . .	47
5.5	The fault-specific negative model and the baseline solution for combined PNF pattern. . . . .	47
5.6	The fault-specific negative model and the baseline solution for all exercises cumulatively. . . . .	47
5.7	The gesture-specific negative model and the baseline solution for shoulder flexion. . . . .	48
5.8	The gesture-specific negative model and the baseline solution for shoulder abduction. . . . .	48
5.9	The gesture-specific negative model and the baseline solution for external rotation. . . . .	48
5.10	The gesture-specific negative model and the baseline solution for elbow flexion and extension. . . . .	48
5.11	The gesture-specific negative model and the baseline solution for combined PNF pattern. . . . .	48
5.12	The gesture-specific negative model and the baseline solution for all exercises cumulatively. . . . .	48

# Chapter 1

## Introduction

### 1.1 Motivation and Scope

Cerebral Palsy (CP) is a neurological disorder caused by a non-progressive brain injury or malformation that occurs while the child's brain is under development. CP affects body movement, muscle control, muscle coordination, muscle tone, reflex, posture, balance and cognitive skills. In most cases, it impacts fine motor skills, gross motor skills, and sensory skills. The effects of Cerebral Palsy are long-term, not temporary. The injury and damage to the brain are permanent. The brain does not heal as the way other parts of the body might. On the other hand, associative conditions may improve over time. Rehabilitation, which includes physical and/or occupational therapy, is among the main intervention methods to promote, maintain and restore the physical well-being of CP patients.

There are various approaches to CP rehabilitation. Virtual reality (VR) based rehabilitation is one of the current approaches. VR-based approaches mainly aim children both because it is effective to perform these exercises in the early ages and because they are often put into practice as serious games to make them more attractive and less boring for children. The emergence of depth cameras to be used in schools and homes made it possible to capture the movements of the patients

and promoted the use of these type of cameras in virtual rehabilitation [5]. However, for CP patients, exercising games targeting the general population proved problematic in some aspects. First of all, these patients sometimes lack the ability to perform some moves properly and complete the game. They may also have cognitive disabilities that cause them to perform unrelated/undefined moves and sometimes requires therapists to step in, which causes the game engine to try and recognize these movements that are out of context. Thus, playing/performing regular exercising games are not practical in this case.

Another problem is the compensation mistakes that are made by the patients during exercises. When the patients have insufficient muscle strength or muscle control, they try to complete the movement by using some other muscles and/or joints, e.g., twisting, bending their elbows during a shoulder exercise. These incorrect exercises are definitely not desired by the therapists.

The solution we provide for the described problems is to develop a gesture recognition system that is designed specifically for cerebral palsy patients. First of all, it should be able to distinguish the movements that are not related to the content of the application from the defined exercises. Secondly, it should be able to detect and capture the compensation mistakes that are done by the patients. Hence, the scope of this work to provide a gesture recognition solution to be used in virtual rehabilitation applications for children with cerebral palsy.

## 1.2 Contributions

The two contributions of this thesis are as follows.

- We propose two alternative methods, called *universal negative model* and *universal positive model*, which enable the detection of non-gesture patterns by producing an adaptive threshold value.
- We examine the problem of detecting small mistakes made by the patients to compensate their lack of ability, control, and strength, and devise a new approach in order to enhance the gesture recognition accuracy in such a case.

## 1.3 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 gives some needed background information on gesture recognition, occupational therapy, and cerebral palsy. In Chapter 3 we describe the base structure of our solution. Chapter 4 focuses on detecting non-gesture patterns and Chapter 5 is mainly about recognizing compensation mistakes. Chapter 6 presents the experimental method and results. Chapter 7 concludes and describes future research possibilities.

# Chapter 2

## Background and Related Work

We elaborate on the related work on gesture recognition approaches and occupational therapy exercises, specifically for Cerebral Palsy.

### 2.1 Gesture Recognition

Gesture recognition is among the fundamental research areas related to human-computer interaction. It is concerned with recognizing meaningful expressions of motion by a human. These motions may include hands, arms, head and/or body. The applications of gesture recognition are manifold [6]:

- medical rehabilitation (e.g., physiotherapy, occupational therapy),
- human activity recognition,
- sign language recognition,
- virtual reality,
- forensic identification, and
- lie detection.



Over the years, various methods have been used for gesture recognition. However, Support Vector Machines (SVMs), Dynamic Time Warping (DTW), Adaboost, and Hidden Markov Models (HMMs) are the ones that are most commonly used in the works that are similar to ours [7].

Regarding SVMs, it should be pointed out that, regular SVMs are capable of classifying a total of two classes, causing the researchers to use multiclass SVMs when more than two gestures are present in the vocabulary. It is stated [8] that approaches that use SVMs perform gesture recognition by classifying single frame gestures or poses, not temporal data. One instance that multiclass SVMs are used together with Kinect is by Biswas et al. [9]. In their study, gestures are classified using only single frames and histograms of depth values in that frame. A similar research in [10] also makes use of single frames when recognizing gestures by SVM.

Bloom et al. [11] use AdaBoost in an unconventional set of gestures, *gaming action dataset*. They state that they choose AdaBoost specifically for the problem at hand. Nevertheless, compared to the other gesture recognition methods, this approach shows a relatively poor performance because of its unique set of gestures.

DTW is another method that gives successful results on gesture recognition. The approach of Sempena et al. [12] is one example of DTW used with depth data for gesture recognition. They claim to have a high success rate. However, they mainly used the method for recognizing repetitive and simple human activities like running and waving. An advantage of DTW is that it is a time-invariant algorithm.

We choose HMMs as gesture recognition approach in our study. Before moving onto examining the method in detail, the reasoning behind this choice needs to be explained. It is clear that comparing studies that use different approaches does not give accurate information because of the differences in the datasets used in these studies. Because of that, research that compares different algorithms using the same set of gestures are inspected thoroughly. Bicego et al. [13] compare 14 different approaches to recognition and the data presented shows that the

most successful methods are HMM-based methods. Another survey by Suarez et al. [14] emphasize the high classification rate and prevalence of HMM-based solutions in gesture recognition. HMMs are also extensively studied in sign language recognition, which is similar to exercise recognition in principle. Comparing the HMM solutions with alternatives in sign language recognition also pictures HMM as a successful approach.

It should be noted that HMM is not argued to be the best approach for gesture recognition. In essence, our assertion is that considering the literature that shows HMM as a successful and relatively consistent model [8] for gesture recognition that can be implemented under certain conditions as a time-invariant method [15], e.g., DTW by taking advantage of the Viterbi algorithm, we conclude that HMM is a gesture recognition method that we can successfully apply to the problem at hand.

### 2.1.1 Hidden Markov Models

An HMM is a doubly-stochastic process with an underlying stochastic process that is not observable (it is hidden), but it can only be observed through another set of stochastic processes that produce the sequence of observed symbols. The observation symbols could be discrete like a coin toss or continuous like speech, gesture, and so on [16].

In order to explain the concept of the HMM, we give a simple coin toss example. Consider yourself in a room with a curtain in front of you. On the other side of the curtain, a coin toss process is performed and you can only see the result of it as heads (H) or tails (T). H and T are the observation symbols. The underlying mechanism that produces H or T is not known to you, which is composed of hidden states.

The model in Figure 2.1 demonstrates how an HMM can be constructed to represent such a coin toss experiment. In the model there are two states, each representing a single coin; one is fair and the other one is biased towards H.

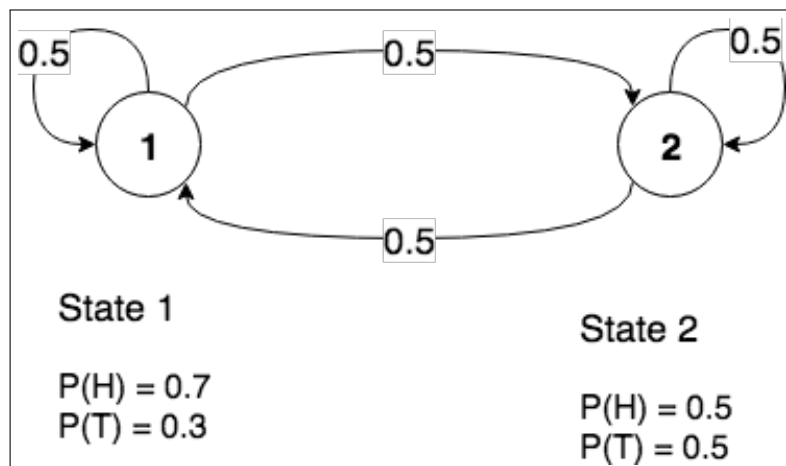


Figure 2.1: A model that corresponds to the tossing of two different coins.

Again, these are the symbols that you can observe. The model has 0.5 transition probability between these two states with the possibility to stay in the same state. It can be assumed that another random process (e.g., another fair coin toss) is used to decide which coin to flip each time and also another one to decide which state to begin with at the start of the experiment. When the state is chosen the result obtained is the one you see behind the curtain. The model notation for a discrete observation HMM is given in Table 2.1.

Table 2.1: The summary of notations.

Notations	Explanations
$\mathcal{T}$	the length of the observation sequence
$\mathcal{N}$	the number of states in the model $S = \{S_1, S_2, \dots, S_{\mathcal{N}}\}$
$\mathcal{M}$	the number of observation symbols $V = \{V_1, V_2, \dots, V_{\mathcal{M}}\}$
$\mathcal{A} = [a_{ij}]$	the state transition probability distribution
$\mathcal{B} = [b_j(m)]$	the observation symbol probability distribution
$\Pi = [\pi_i]$	the initial state distribution

Using the model, an observation sequence,  $O = O_1, O_2, \dots, O_{\mathcal{T}}$ , is generated as follows:

1. Choose an initial state  $q_1$  according to  $\Pi$
2. Set  $t = 1$
3. Choose  $O_t$  according to  $b_{q_t}(m)$ , the symbol probability distribution
4. Choose  $q_{t+1}$  according to  $a_{q_t q_{t+1}}$
5. Set  $t = t + 1$ ; return to step 3 if  $t < \mathcal{T}$ ; otherwise, terminate the procedure.

We use the compact notation  $\Lambda = (\mathcal{A}, \mathcal{B}, \Pi)$  to represent an HMM. The specification of an HMM involves the choice of the number of states,  $\mathcal{N}$ , the number of discrete symbols  $\mathcal{M}$ , and the specification of the three probability densities  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\Pi$ .

### 2.1.1.1 The Three Problems for HMMs

In order for HMMs to be useful in real-world applications, there are three fundamental problems that need to be addressed [17]. These problems are as follows:

- *Problem 1:* Given the observation sequence  $O = O_1, O_2, \dots, O_{\mathcal{T}}$  and the model  $\Lambda = (\mathcal{A}, \mathcal{B}, \Pi)$ , how do we compute  $P(O \mid \Lambda)$ .
- *Problem 2:* Given the observation sequence  $O = O_1, O_2, \dots, O_{\mathcal{T}}$ , how do we determine an optimal state sequence  $Q = q_1, q_2, \dots, q_{\mathcal{T}}$ .
- *Problem 3:* How do we select the appropriate model parameters  $\Lambda = (\mathcal{A}, \mathcal{B}, \Pi)$  to maximize the value of  $P(O \mid \Lambda)$ .

Problem 1 is the problem that we solve to classify the given observation sequence. We do this by computing the probability of the model producing the

sequence of observations at hand. The answer to this problem and the computed probability can be viewed as the score of the model. If we have the solution to Problem 1, we can compute the probability of producing the observation sequence, compare each model based on their score and choose the model with the best score.

In Problem 2, which is a typical estimation problem, the aim is to specify a hidden state sequence that is optimal for the observation sequence. Nevertheless, there are various possible optimality criteria that could be utilized to solve this problem and as a result, the choice of the criterion has significant effects on the results. The recovered hidden state sequence is typically used to examine the model structure and get the intended information such as the behavior of the individual states.

Problem 1 is the testing problem for our model and Problem 3 is actually the one that should be handled beforehand because it is the training problem. The way Problem 3 is solved in an application of HMM, actually defines the success of the application. In this stage, we try to find out the best possible parameters that optimally adapt the model to the provided training sequences.

### 2.1.1.2 Solutions to the Three HMM Problems

When an observation sequence  $O = O_1, O_2, \dots, O_{\mathcal{T}}$  is given, there can be NT different state sequence  $Q = q_1, q_2, \dots, q_{\mathcal{T}}$ . In that case,  $P(O | \Lambda)$  can be computed as follows [18]:

$$P(O | \Lambda) = \sum P(O, Q | \Lambda), \quad (2.1)$$

which is clearly not a computationally-efficient solution. This procedure is called the forward-backward procedure and it is used in HMM computations. It is an iterative procedure and makes use of the dynamic programming principles. The forward procedure in simple terms is as follows [18]:

The initiation of the forward probabilities with the joint probability of  $q_i$  and initial observation  $O_1$ .

$$\alpha_1(i) = P(o_1 | q_1 = S_i, \Lambda)P(q_1 = S_i) \quad (2.2)$$

$$\alpha_1(i) = \pi_i b_i(o_1), \quad t = 1, \quad i = 1, 2, \dots, N \quad (2.3)$$

The next iteration is computed as

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), j = 1, 2, \dots, N \quad (2.4)$$

The desired calculation of  $P(O | \Lambda)$ :

$$P(O | \Lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.5)$$

The backward procedure is as follows [18]:

$$\beta_{T-1}(i) = \sum_{j=1}^N a_{ij} b_j(o_T), \quad i = 1, 2, \dots, N \quad (2.6)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad i = 1, 2, \dots, N \quad (2.7)$$

$$P(O | \Lambda) = \sum_{i=1}^N \beta_1(i) \pi_i b_i(o_1) \quad (2.8)$$

As we pointed out, there are various choices for the optimality criterion that will be used when solving Problem 2. Hence, there are different ways to solve the problem. One possible approach is to evaluate each state individually and choose

the most likely ones. This approach provides us with the maximized set of correct states. However, this computation has a fundamental problem; this procedure does not take into account the probability of state transitions that cannot be made between states  $i$  and  $j$  ( $a_{ij} = 0$ ). Because of that the Viterbi algorithm is used to overcome this problem [19]. The Viterbi algorithm is very similar to the forward-backward procedure; the difference is when considering the state at time  $t$ , instead of taking into account of all possible preceding states, the Viterbi algorithm only uses the state with the highest probability.

Finally, in order to solve problem three, which is the training problem, an expectation maximization process, the Baum-Welch algorithm is used [17]. Another approach to this problem is the Viterbi Path Counting method [20]. However, the Baum-Welch algorithm is much more studied and common in the literature and we selected it because of its popularity.

### 2.1.1.3 Types of Hidden Markov Models

There are various options when it comes to designing the model for the HMM approach [18]. The most popular three types of HMMs are as follows:

- *Ergodic model*: A model in which it is possible to reach any state from any other state (see Figure 2.2).
- *Left-to-right model*: A model in which a state can only be reached from the preceding states. These types of models inherently impose a temporal order and thus widely used in speech and gesture recognition (see Figure 2.3).
- *Parallel left-to-right model*: Similar to the Left-to-Right model, except that it has several paths through the states (see Figure 2.4).

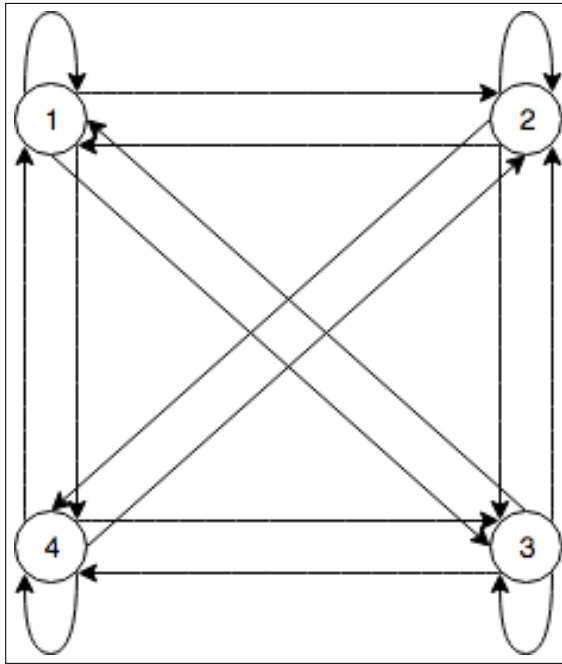


Figure 2.2: Ergodic Hidden Markov Model

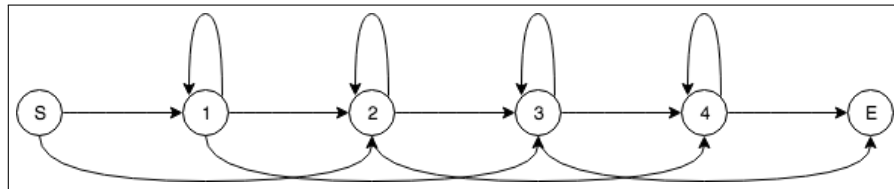


Figure 2.3: Left-to-Right Hidden Markov Model

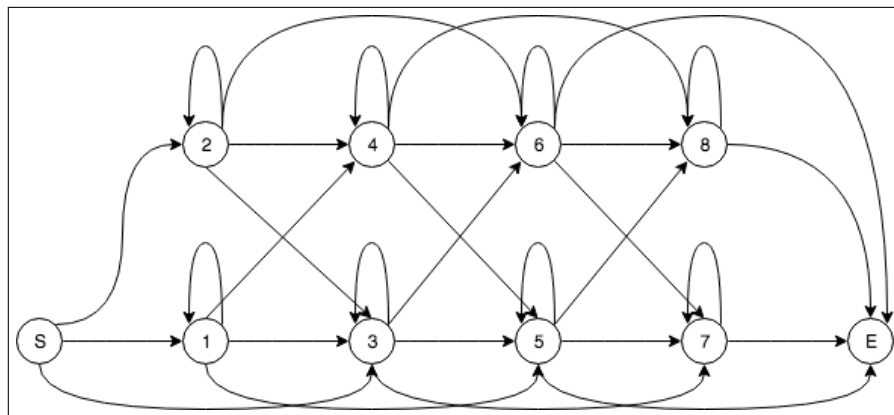


Figure 2.4: Parallel Left-to-Right Hidden Markov Model



#### 2.1.1.4 Literature on Using Hidden Markov Models

To the best of our knowledge, even though there is not any research that concentrates on the recognition of the erroneous exercises practiced by the patients as we do in this study, there are many pieces of research on human activity and/or exercise recognition with HMM.

Lu et al. [1] propose an HMM-based method using Kinect RGB-D camera. Figure 2.5 gives the overview of their method. They extract the joint information using the depth data provided by the camera and then generate histogram data of joint locations. (with the spherical coordinate system) In order to overcome the continuous observation symbol problem, they utilize vector quantization which makes it possible to use discrete HMM. They tested their proposed approach with their own dataset and also used the dataset MSR Action 3D provided by Microsoft to compare their performance with other approaches. It is claimed in the work that the proposed method outperforms the works that have been carried out before their publication.

Yang et al. [21] focus on hand gesture recognition primarily and thus involves the segmentation of hand from RGB data before HMM. They do not use depth cameras. The features selected for recognition are hand position, velocity, size, and shape. Another problem they have dealt with is data aligning problem. It is mainly the time-variance problem and the method they utilized is a simple aligning algorithm. It is asserted that with the use of various features together in recognition, they managed to increase the recognition performance.

Uddin et al. [2, 22] propose an HMM-based approach that also uses histogram data. However, both silhouette and joint data are determined as features for different setups and a comparison is made between them. It is pointed out that their approach gives better results using joint-based features. It is also important that joint angles and not locations used in this research and judging by the experimental results it is one of the better approaches on gesture recognition with HMM. Type of HMM in this work is the Left-to-Right Model because of its temporal nature. The problem requirements in these researches are similar to

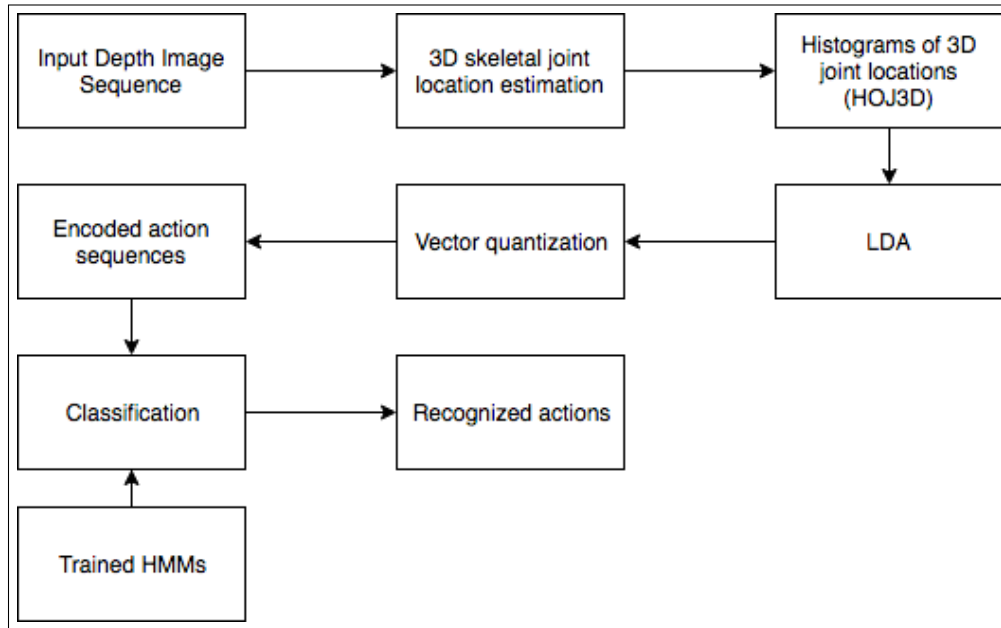


Figure 2.5: The overview of the method proposed in [1].

ours in this thesis and their methodology provides a good baseline approach for dealing with gesture recognition. The framework of their recognition system is presented in Figure 2.6.

## 2.2 Occupational Therapy

We focus on the recognition of occupational therapy exercises. Hence, having a better knowledge of the domain of occupational therapy together with its goals and application areas is significant in order to have a complete understanding of the work presented.

In simple terms, occupational therapy is a sub-branch of physiotherapy, that focuses on the daily activities of the patients. Even though occupational therapy practitioners use similar exercises for the rehabilitation of the patient, in terms of context and the evaluation of these exercises, it has different characteristics.

According to the practice framework published by The American Occupational

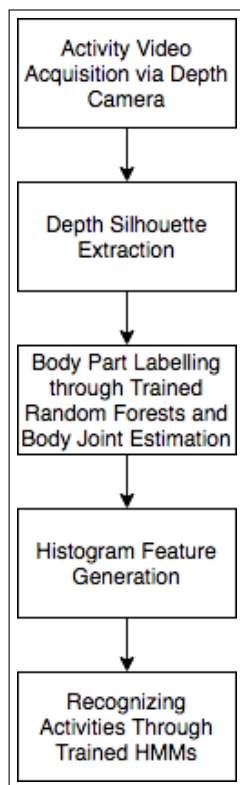


Figure 2.6: The framework of the recognition system proposed in [2].

Therapy Association; occupational therapy aims to enhance the daily lives of individuals and groups in homes, schools, workplaces, and so on by utilizing the everyday life activities in the therapy. Occupational therapists provide development in body functions, body structures, motor skills, processing, and social interaction skills by putting their knowledge of the transactional relationship among the person and occupations the person is engaged into use and creating an occupation-based intervention plan with the aim of successful social participation. Because occupational therapists aim the end result of participation, when needed, they manage and modify the environment and objects within to increase engagement of the person. Habilitation, rehabilitation, promotion of health and wellness for people with needs related or unrelated to their disability are the objectives of occupational therapy services [23].

Rehabilitation exercises are one of the tools that are being used by occupational

therapy practitioners in various different cases. The target audience for occupational therapy includes all age groups from children to seniors and many different types of disorders, namely: cerebral palsy, stroke, and Parkinson's among many others [24].

Taking advantage of the latest technologies is not uncommon in occupational therapy. Especially in recent years, using virtual therapy and/or augmented reality applications in therapy sessions gained traction. Wentao et al. [25] use robotic therapy practices for cerebral palsy patients. In order to train the virtual therapist (robot), they use HMM as a pattern recognition method.

However, with the emergence of RGB-D cameras that are more commercially available, gesture recognition based on data obtained from RGB-D cameras became one of the main focus areas [5]. One can find extensive research on the subject of rehabilitation with gesture recognition that is done in cooperation with therapists. These works are mainly on the subject of how these new types of rehabilitation practices benefit the patients.

Chang et al. [26] propose a Kinect-based upper limb rehabilitation system for cerebral palsy patients. Based on their experimental work, they argue that the system they devised managed to increase the motivation of the test subjects and also facilitated an improvement on the success-rate of the exercises. In another publication, Chang et al. [27] test a similar system on young adults and get similar successful results with regards to the rehabilitation of patients.

Another research by Pedraza et al. [28] describes a Kinect-based virtual reality system and claims to improve patient mobility, aerobic capacity, strength, coordination, and flexibility. Examining the existing research on this area shows that using pattern recognition for occupational therapy has the potential to produce beneficial results for patients.

### 2.2.1 Cerebral Palsy

Cerebral Palsy (CP) is a well-studied neurologic condition beginning in early childhood when the child’s brain is still developing. The condition is non-progressive and persistive through the lifespan. The term CP includes several different types of disorders caused by disturbances in the developing brain which results in activity, mobility, sensation and cognition problems [29].

CP could be classified according to the way motor skills are affected which also indicates which part of the brain is damaged: spastic, dyskinetic or ataxic. Another type of classification is dictated by the location of impairment. Quadriplegia is when both arms and legs are affected, diplegia defines the patients with impairment at both legs and hemiplegia is when one arm and one leg on the same body side is affected, as a result of brain damage that affects one hemisphere [30, 31].

Gross Motor Function Classification System (GMFCS) and Manual Ability Classification System (MACS) are two classification standards that are used to differentiate patients according to the severity of impairment. We focused on children that need the occupational therapy exercises to improve their ability to complete their daily activities and also able or expected to be able to perform the exercises correctly. Another requirement was the ability to stand because otherwise, the tracking accuracy of the depth camera reduces dramatically. As a result, children that are classified in level 1 or 2 of GMFCS and in level 2 or 3 of MACS are targeted during our study.

### 2.2.2 Exercise Set

The depth camera shows better performance when capturing the upper extremities and the patients often need to sit, lay down or hold onto something when performing the lower extremity exercises which also reduces the camera’s accuracy. Because of these reasons, the chosen exercises for the purpose of this thesis are upper extremity functional exercises. These moves focus on the movement

of arms, specifically shoulder and elbow joints. These exercise groups are considered important by the occupational therapist because the better use of upper extremities affects the daily lives of the patient greatly.

From the group of upper extremity functional exercises, five main gestures were selected: shoulder flexion ( $180^\circ$ ), shoulder abduction ( $90^\circ$ ), shoulder external rotation, elbow flexion and extension combined and PNF pattern of all other four movements combined. The features determined for each exercise is given below.

- *Shoulder flexion*: the shoulder angle on the X-Z plane, the shoulder angle on the Y-Z plane, the elbow angle on the X-Z plane, the elbow angle on the Y-Z plane, and the body angle on Y-Z plane.
- *Shoulder abduction*: the shoulder angle on the X-Y plane, the shoulder angle on the X-Z plane, the elbow angle on the X-Y plane, the elbow angle on the X-Z plane, the head angle on the X-Y plane, and the body angle on the X-Y plane.
- *External rotation*: the shoulder angle on the X-Y plane, the shoulder angle on the X-Z plane, the elbow angle on the X-Y plane, the elbow angle on the X-Z plane.
- *Elbow flexion-extension*: the shoulder angle on the X-Y plane, the shoulder angle on the X-Z plane, the elbow angle on the X-Y plane, the elbow angle on the X-Z plane, the head angle on the X-Y plane, and the body angle on the X-Y plane.
- *Combined PNF pattern*: the shoulder angle on the X-Y plane, the shoulder angle on the X-Z plane, the elbow angle on the X-Y plane, the elbow angle on the X-Z plane, the head angle on the X-Y plane, and the body angle on the Y-Z plane.

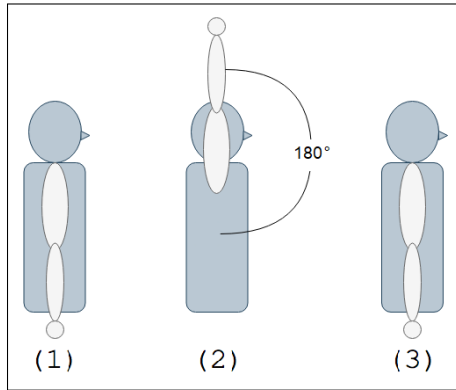


Figure 2.7: Shoulder flexion: starting position (1), arm is raised  $180^\circ$  from the front while keeping the elbow angle as  $180^\circ$  (2), and the end position (3).

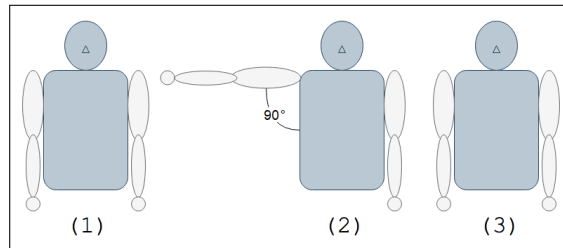


Figure 2.8: Shoulder abduction: starting position (1), arm is raised  $90^\circ$  from the side while keeping the elbow angle as  $180^\circ$  (2), and the end position (3).

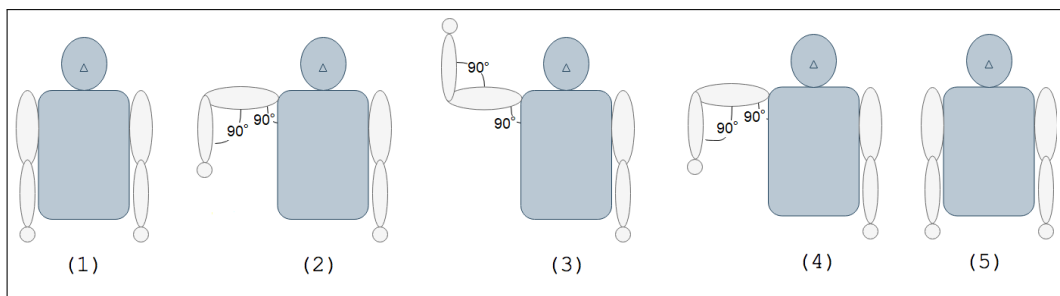


Figure 2.9: External rotation: starting position (1), arm is raised from the side while keeping the elbow and shoulder angles as  $90^\circ$  (2), the arm is rotated using only the shoulder joint (3), the arm is rotated back (4), and end position (5).

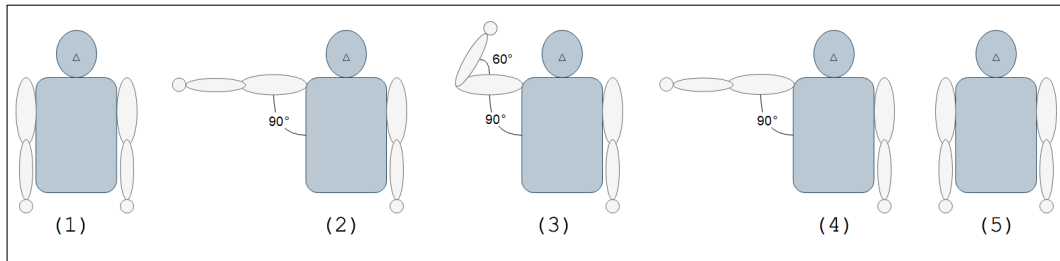


Figure 2.10: Elbow flexion and extension: starting position (1), the arm is raised to the side while keeping the elbow straight and the shoulder angle as  $90^\circ$  (2), the elbow is flexed to  $60^\circ$  (3), the elbow is extended back (4), and the end position (5).

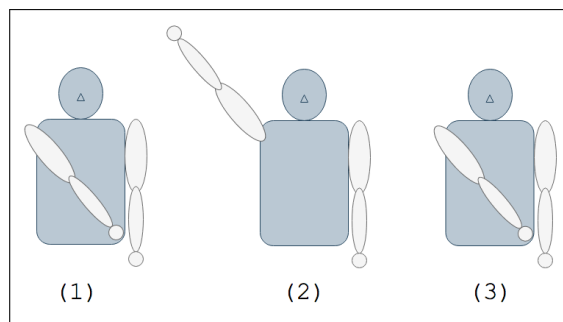


Figure 2.11: Combined PNF pattern: starting position (1), arm is raised diagonally while keeping the diagonal movement line straight (2), and back to the end position (3).





Figure 2.12: Kinect v1.

## 2.3 Depth Cameras

Microsoft Kinect® is developed by Microsoft and PrimeSense under the project name “Project Natal” as a consumer-grade RGB-D camera [32]. Kinect v1 is first released in November 2010 and Kinect for Windows SDK is released in 2012 [32]. Even though Kinect was released as a game controller for Xbox, it is used in a variety of areas. In addition to Microsoft SDK, some other open source APIs are also released like OpenKinect and OpenNI. The emergence of Kinect originated numerous studies and developers to work on projects/researches that utilize Kinect.

Kinect mainly provides the depth data along with the RGB data. However, together with the Microsoft Software Development Kit (SDK) or any other open-source Application Programmer’s Interface (API), the body silhouette and the skeleton data (joint coordinates) can be obtained.

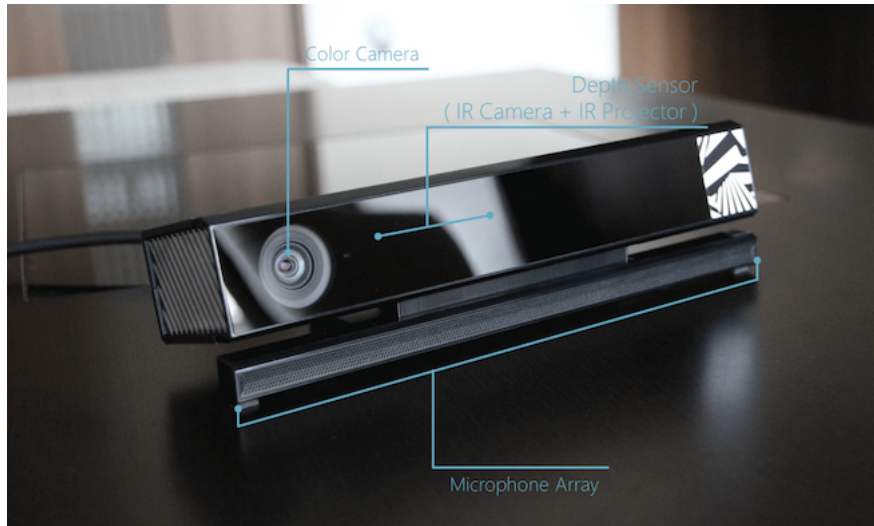


Figure 2.13: Kinect v2.

### 2.3.1 Kinect v1 and Kinect v2

So far, two versions of Kinect are present: v1 and v2. While the first version makes use of the structured light method, version 2 utilizes the time-of-flight technology. It is claimed that v2 generates better results than v1, both by the producer (Microsoft) and by the research community, e.g., [33]. Table 2.2 provides a technical comparison of the features of Kinect v1 and v2.

Table 2.2: The comparison of Kinect v1 and v2.

	<b>Kinect v1</b>	<b>Kinect v2</b>
RGB Camera (pixel)	640 × 480 or 1280 × 1024	1920 × 1080
Depth camera (pixel)	640 × 480	512 × 424
Max depth distance (m)	4.0	4.5
Min depth distance (m)	0.8	0.5
Horizontal FOV (degrees)	57	70
Vertical FOV (degrees)	43	60
Skeleton joint defined	20	26
Full skeleton tracking	2	6

### 2.3.2 Kinect in Rehabilitation

Kinect has been used in many different areas, such as gaming, 3D reconstruction, and motion capture. However, what we are concerned about here is the use of Kinect in rehabilitation and exercise recognition. The research shows that it is used in many different disorders such as Parkinson's, Cerebral Palsy, and stroke patients.

Clark et al. [34] examine the validity of Kinect for postural control assessment and compare it against marker-based 3D motion capture cameras that provide precise measurements. They conducted tests of functional reach and timed standing balance methods. Their results show that Kinect is able to provide reliable data when compared to professional motion capture systems. They conclude that Kinect can be used in clinical screening programs for a wide range of patient populations.

The research presented in [35] analyses the reliability and validity of Kinect in functional assessment activities, which we are also concerned with, and compares it with stereophotogrammetry methods. They designed tests for four different joints: shoulder, elbow, hip and knee and the results presented suggests that in lower body exercises it is determined that Kinect behaves relatively poorly but presents reliable results in upper-limb exercises compared to stereophotogrammetry technology. These results confirm our choice of using Kinect and determining upper-body rehabilitation activities in our dataset.

In [3] and [36], Kinect is assessed for use for the rehabilitation of people with Parkinson's disease and stroke disorders, respectively. Their work also shows that Kinect provides significant results compared to the professional systems and are reliable to use in a rehabilitation context. A comparison of Kinect data and a research-grade motion capture system OptiTrack can be seen in Figure 2.14.

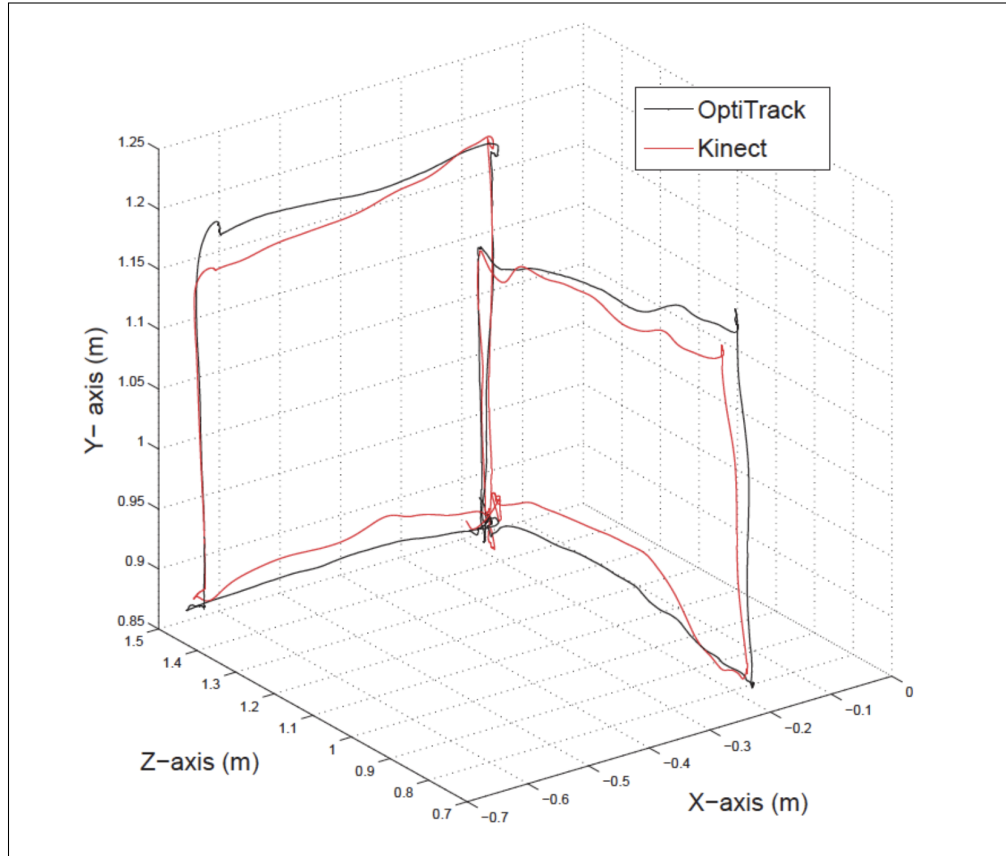


Figure 2.14: The comparison of the OptiTrack motion capture system and the coordinate data provided by Kinect [3].

## 2.4 Definition of the Setup

Our aim is to propose an exercise recognition system that specifically targets the requirements of the occupational therapists. In our setup, when a cerebral palsy patient is required to perform a specific exercise, the system should be able to

- eliminate the unrelated gestures that are done by the subject during the exercise session and not count them as correct or incorrect exercises and
- recognize the compensation mistakes of that specific gesture and count them as incorrect exercises.

The goal is not to distinguish exercises from each other, but to distinguish the

compensation mistakes for the specified exercise. Hence, each time the patients are directed to perform one predetermined exercise and during the session all the gestures performed are either: non-gestures (gestures that are defined in our exercise set but not the predetermined exercise are also classified as non-gestures), the correct version of the predetermined exercise, or incorrect version of the predetermined exercise.

Tables 2.3 and 2.4 show how many times each patient performed the gestures for validation and training set, respectively. In the training set, the number of incorrect exercises are much higher, that is because there are *five* different compensation mistakes defined for each exercise. The correctness of each gesture is determined by the supervising occupational therapists.

The patients that performed the exercises are chosen so that they are able to perform the exercises without direct physical assistance so that Kinect can provide more stable data and also abrupt movement changes like stopping, accelerating or decelerating during the exercise could be prevented. It should also be noted that when recording the gestures a basic normalization is done, i.e., we fixed the number of frames recorded for each gesture by sampling the frames.

Table 2.3: The validation set. The columns are Shoulder Flexion (SF), Shoulder Abduction (SA), External Rotation (ER), Elbow Flexion and Extension (EFE), combined PNF pattern (PNF), and Nongesture (NG). T and F stand for the correct and incorrect versions of each exercise, respectively.

<b>Patients</b>	<b>SF</b> <b>T/F</b>	<b>SA</b> <b>T/F</b>	<b>ET</b> <b>T/F</b>	<b>EFE</b> <b>T/F</b>	<b>PNF</b> <b>T/F</b>	<b>NG</b>
Patient #1	5/5	5/5	5/5	5/5	5/5	10
Patient #2	5/5	5/5	5/5	5/5	5/5	10
Patient #3	5/5	5/5	5/5	5/5	5/5	10
Patient #4	5/5	5/5	5/5	5/5	5/5	10
Patient #5	5/5	5/5	5/5	5/5	5/5	10
Patient #6	5/5	5/5	5/5	5/5	5/5	10
Total	30/30	30/30	30/30	30/30	30/30	60

Table 2.4: The training set. The columns are Shoulder Flexion (SF), Shoulder Abduction (SA), External Rotation (ER), Elbow Flexion and Extension (EFE), combined PNF pattern (PNF), and Nongesture (NG). T and F stand for the correct and incorrect versions of each exercise, respectively. Because the training set for nongestures are not performed by the six patients, the NG values for them are left blank.

<b>Patients</b>	<b>SF</b> <b>T/F</b>	<b>SA</b> <b>T/F</b>	<b>ET</b> <b>T/F</b>	<b>EFE</b> <b>T/F</b>	<b>PNF</b> <b>T/F</b>	<b>NG</b>
Patient #1	30/150	30/150	30/150	30/150	30/150	
Patient #2	30/150	30/150	30/150	30/150	30/150	
Patient #3	30/150	30/150	30/150	30/150	30/150	
Patient #4	30/150	30/150	30/150	30/150	30/150	
Patient #5	30/150	30/150	30/150	30/150	30/150	
Patient #6	30/150	30/150	30/150	30/150	30/150	
Total	150/900	150/900	150/900	150/900	150/900	1120

## Chapter 3

# Gesture Recognition Model

In this chapter, we describe our baseline method for gesture/exercise recognition. The focus of the design will be the unique characteristics of the problem at hand and the resulting solution is intended to have the ability to recognize and differentiate different upper-body exercises. Various problems regarding feature selection, model structure, scaling problem, and continuous observation symbols are addressed in detail and the solutions are elaborated in depth.

It is important to note the different requirements of the problem and not to suggest a classical activity recognition approach here. It is specifically emphasized in [11] that when the recognition problem has untraditional aspects and the devised system does not provide tailored solutions to these, experiments can present diminished results in terms of recognition accuracy. The framework of the proposed solution is given in Figure 3.1.

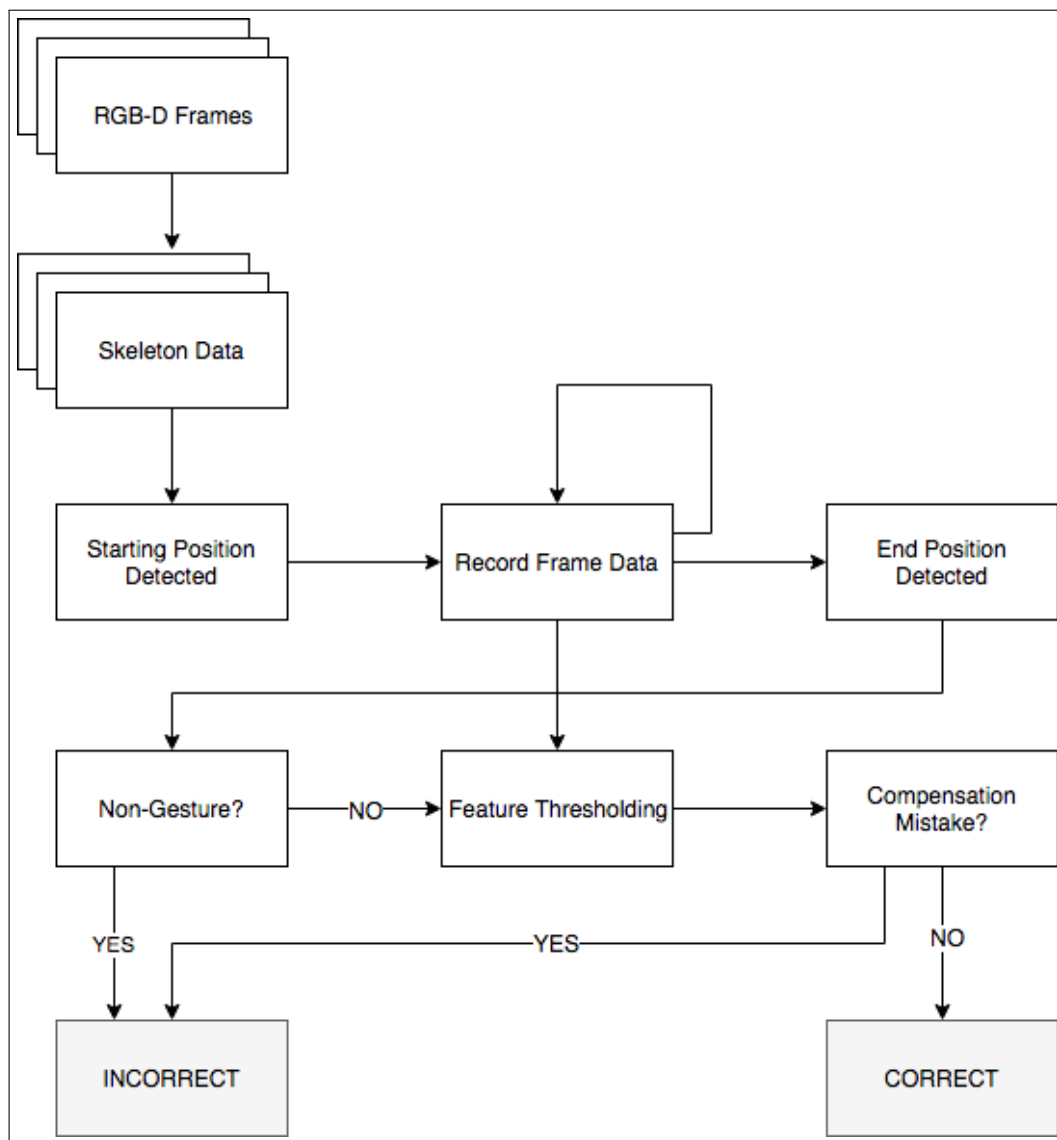


Figure 3.1: The framework of the proposed solution



Our framework first takes the frames from the depth camera as input and processes the skeleton data provided by the camera’s SDK. When the starting position is detected, we record the data at each frame as the gesture data until we detect the end position. Afterward, we process the gesture data using the universal negative model to see if it is a non-gesture (see Chapter 4). If the gesture is defined in the exercise set, we apply feature thresholding to the collected data (see Chapter 5). We then process the manipulated data using different negative models to detect compensation mistakes (see Chapter 5). Finally, if no mistake is detected, we classify the gesture as a correct gesture.

### 3.1 Feature Selection

Feature selection for the recognition task is one of the most significant steps in gesture recognition. Some studies used various sets of features within the same system and the results are drastically different between them [22].

The first thing to determine for feature selection is the data source for our approach. Because Kinect RGB-D camera is used as hardware, we have two different sources of data: silhouette and skeleton (joint) data.

The silhouette data is widely used in activity recognition systems. However, compared to joint data, use of HMM with silhouette data is relatively uncommon. In [37], activity data is classified using nearest neighbor matching. The recognition features and body shape and gait position during walking activity. Zhang et al. [38] utilize a Bag-of-3D points approach for recognition. This is another example of non-HMM gesture recognition. Bobick et al. [39] propose a successful recognition approach that uses modified silhouette data with a non-HMM method. As stated before, the research in [22] uses silhouette with HMM but the skeleton data shows up to 84% performance gain. Hence, it can be concluded that the silhouette data is not suitable for use with Hidden Markov Models whereas other approaches give better results. Further inspection makes it evident that recognition using silhouette data generally used for daily activity recognition

instead of gestures like rehabilitation exercises. These findings suggest that the use of joint data is more suitable to our problem at hand.

MS Kinect provides joint data in 3D space. The joint data also has its uses in the literature on gesture recognition. Xia et al. [1] examine various different approaches with joint data. It is possible to utilize joint locations, joint motions and/or joint angles as features. Campbell et al. [15] examine the advantage and disadvantages of each type of joint data. They focused on features' shift-invariant and rotation-invariant properties. It is argued that when joint locations are used, the approach becomes vulnerable to expected coordination shifts in 3D space and also the rotations of the subjects. When joint angles are used, the system becomes shift/coordinate-invariant, however, it is still affected by rotations in space. Thus, they propose the use of joint motion (i.e., derivative of location or angle) as a shift-invariant and rotation-invariant feature set. However, one disadvantage of using derivatives is that it depicts the same gestures performed in different speeds as different gestures, naturally. We can also say that it causes the loss of speed-invariance property.

Coordinate shifts are important in our problem because the position of the subject relative to the camera is not always the same. Also, the body metrics of each patient is also different. This makes it appropriate to choose a shift-invariant feature set. However, rotation-invariance is not needed in this case. Because depending on the gesture in our dataset, the subject directly faces the camera or stands perpendicular to it during training and recognition phases. These shreds of evidence suggest that using joint angles have no disadvantages with regards to shift and rotational variance in our case and also provides the Viterbi algorithm with time warping behavior [15].

Another requirement for joint angle feature is that it is important for the evaluation of occupational therapy exercises because the correctness of each gesture is generally decided by the joint angles [24]. As a result, we chose different joint angles or their 2D projections for each different gesture in our dataset. For each gesture, the requirements and standards for the gesture are taken into account and the joints that should be tracked for the gesture to be classified accordingly

is determined by the occupational therapy researchers.

### 3.1.1 Feature Count

We choose the features for each gesture with the expertise of an occupational therapist according to the nature of the exercise. However, it is also necessary to determine or limit the number of features as shown in other similar studies.

In order to determine the number of features, a series of test are conducted. In this stage, gestures that are not defined in our exercise set or incorrect exercises are not taken into consideration. The tests are conducted as a multi-class classification problems where we try to distinguish each of the *five* exercises from each other. Each exercise is performed a total of 20 times by three different subjects. These performances are recorded so that for each feature count we can use the same data. The results show that after reaching three as the feature count, we are able to get reasonable results over 0.80 F1 score. Using four, five and six features provide results over 0.91, four feature being the highest at 0.96 F1 score. However, four features could be insufficient when trying to distinguish incorrect gestures from the correct ones. As a result, it is determined that the feature count should be between four and six, even though four is the highest. The graph that shows the test results are depicted in Figure 3.2.

## 3.2 Hidden Markov Model Structure

HMMs have different types and the structure of each type dictates the recognition property of the model devised. Our choice of model type is a Left-to-Right model. It is already stated that the inherent temporal structure of Left-to-Right model makes it useful for recognition problems that have temporal data like gesture and speech recognition.

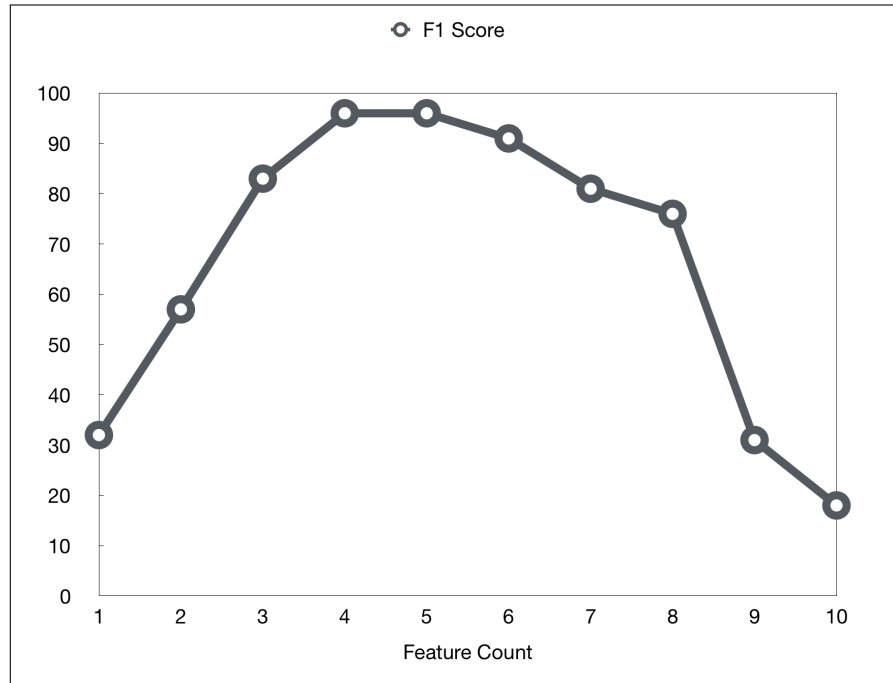


Figure 3.2: The graph depicting the feature count versus F1 score.

The choice of model type is not sufficient to define our model structure. Another point that needs addressing is the number of states. There is no set-in-stone approach for deciding the number of states and states being hidden states are the reason for it. One should consider the properties of the recognition problem concerned and determine the state count accordingly. However, one thing to pay attention is that when the training data size is constant, increasing the state count result in declined performance [17]. Hence, one needs to find the minimum count of states to represent the gesture. Although the states in HMM are not a direct representation of frames or time intervals of the gesture, they tend to produce observation symbols showing similar feature properties. When we analyze our gesture set, each gesture starts with a “resting pose” [15], then the related upper-limb reaches a starting point and reaches the final pose before taking the same route back to resting pose. This process implies four stages of gesture (excluding resting poses at the beginning and end): rest-to-start, start-to-final, final-to-start, and start-to-rest. Thus, we designed our model with four states between one start and one end states, a total of six states. The difference between start and end states is that they have no self-transitions (see Figure 3.3).

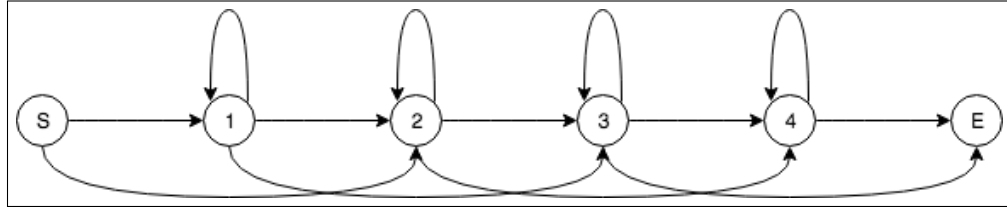


Figure 3.3: The proposed HMM structure.

As it is seen from Figure 3.3 each state can also “skip” the state that supersedes it. These transitions are added to account for possible missing parts/frames of the gesture. The research also shows that HMM has the capability to learn to skip several states entirely, resulting gestures with 4-5-6 state modeled using a 6-state structure [15]. The initial state transition probabilities are uniformly distributed [17].

### 3.3 Scaling Problem

Scaling is a commonly occurring problem in HMM applications. The reason is that the state and observation transition probabilities tend to reach zero geometrically fast. Hence, a scaling technique is needed to avoid mathematical underflow [16]. One basic solution proposed in [17] is to use a scaling coefficient for both transitions. Nonetheless, there is no straightforward method of determining the scaling factor. Another solution provided by Mann [40] uses a numerically stable HMM implementation that also updates all relative computation formulas for the transition probabilities.

The method mainly uses natural logarithms of each probability value instead of their floating point representations. In order to achieve this functionality, we created a variable type called “DoubleLN” and implemented all mathematical operations (summation, multiplication, division, exponential operations, and so on). The functions of HMM, namely the forward procedure, the backward procedure, the Viterbi algorithm, and the Baum-Welch algorithm, are updated as explained in [40].

## 3.4 Continuous Hidden Markov Model

Hidden Markov Models discussed and examined in Chapter 2 are mainly discrete HMM solutions. In discrete HMM, observation symbols are discrete values. Nonetheless, especially in gesture recognition the natural properties of gestures are not discrete but continuous. This case also elaborated in [16] and two different solutions proposed in order to extend discrete HMMs to continuous HMMs. Both solutions involve the introduction of mixture densities, Gaussian M-component mixture densities and Gaussian autoregressive M-component mixture densities. We preferred the latter for our solution.

Another solution that can be applied to continuous observation symbols is using vector quantization. This method removes the need for a continuous HMM and transforms the observation data into discrete values. However, during quantization loss of precision in observation data is inevitable and may cause a decrease in the recognition precision. Thus, we chose to extend the discrete HMM solution to the continuous HMM.

A comprehensive solution presented in [18] with regards to continuous HMMs. The updated computation formulas for observation densities, state transition probabilities and Baum-Welch parameters could be seen in the same study as well.

## Chapter 4

# Detection of Non-gesture Patterns

As expanded on in previous chapters, HMM generates the recognition result by comparing the likelihoods of all trained models and selecting the one with the highest value. This approach works in contexts that all the input data is known to be within the predetermined set of gestures. However, when it is possible to have inputs that are not related to any gesture trained, like studies in [41], [39] and [42], HMM does not function as intended. Even though the input is not in any way related to the dataset, HMM picks the model with the highest probability and naturally, this phenomenon causes problems.

The problem we are dealing with also has a similar context. Despite our predetermined dataset with both correct and incorrect gesture performances, it is always expected from a subject to perform a gesture that is entirely unrelated. Subject walking in/out of Kinect's field of view, resting between exercises, therapists interventions are examples of such situations. Thus, we need to propose solutions to overcome this disadvantage of HMM.

Specifying a constant threshold value does not work because the likelihoods of the models fluctuate altogether depending on the input properties, the length

of the observation sequence being one of them [39, 43]. Therefore, a mechanism should be devised that would produce an adaptive threshold value. This objective could be achieved by another model or models that generate a threshold value based on the input gesture. The ideal threshold value for a correct gesture would be less than that of the corresponding model and would be greater than that of all other models when a non-gesture is given as input. We have two solutions that could provide us with adaptive threshold values that are close to the ideal: *universal negative model* and *universal positive model*. We also compared the performance of our two methods with the performance of the threshold model proposed by Lee and Kim [4].

## 4.1 Universal Negative Model

The universal negative model is the concept of having a trained weak hidden Markov model that encapsulates all gestures that are not included in our dataset. Hereby, it is expected that when the observation sequence is a non-gesture, the model with the maximum likelihood would be the universal negative model. It can also be considered as another model competing with our gesture models with the distinction of representing multiple gestures.

During the training session, we directed a total of five subjects to perform various random gestures in front of Kinect for eight hours. The gestures included possible gestures that can be performed in the subjects' daily lives and in possible recognition scenarios. The length, speed and the number of joints used in gestures were not restricted during the training session. The only restriction was that none of the gestures should be similar to the ones we have in our dataset. A total of 1120 gestures were recorded for the training of the universal negative model. Only one model is used for all of the correct gestures in our dataset.

The model designed for the universal negative model can be seen in Figure 4.1. We used a parallel left-to-right model [17]. It is essentially a left-to-right model that obeys all state transition probabilities of linear left-to-right models. Its



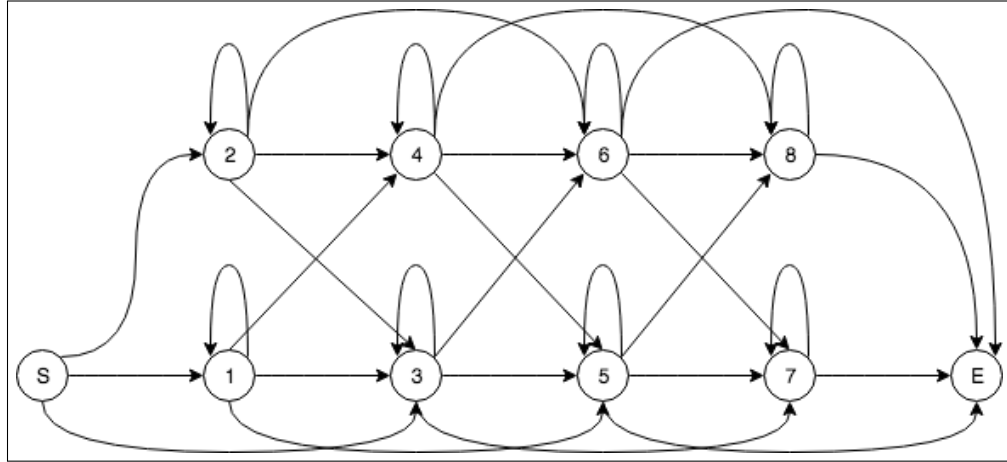


Figure 4.1: The structure of the universal negative model.

difference is that it actually is a cross-coupled connection of two parallel left-to-right models. We decided to use such a model because the gestures in the training set do not have well-defined properties, so it was not possible to generate a linear left-to-right model that fits the general properties. More importantly; because we used many different gestures for the training of this model, even though none of them was part of our original exercise set, the model was generating high likelihoods even for defined gestures. Thus, we needed a more fitting model and the parallel left-to-right structure we implemented with more hidden states delivered the desired results. The intuition behind the choice of a parallel model is to be able model the movements of left and right arm separately.

## 4.2 Universal Positive Model

The universal positive model is a loosely fit model for the superset of all gestures in the dataset. Its premise is to generate a probability that is smaller than that of the correct gesture model for a given gesture, but larger than for a non-gesture motion. In simple terms, one can think of this model as the average model of all other models. Thus, when the gesture given to the system is defined in the dataset as a correct or faulty gesture, it will generate a likelihood value that is less than the corresponding model. When the given input is a non-gesture, because it

is a weak model, it is expected to fit better into the non-gesture than the dataset models. However, if it behaves exactly like an averaging model, it is possible for this approach to generate values that is always less than some model in our dataset.

As stated, for this approach to work; the model should be a loosely-fitted model. Hence, we used the same type of HMM structure that we used for our defined exercise gestures (see Figure 3.3). It should be noted that we didn't use just one model for all types of gestures. Each gesture is recognized using one correct and several faulty gesture models, i.e., negative models (see Chapter 5). We trained a separate universal positive model for each separate gesture.

### 4.3 Threshold Model

The threshold model is proposed by Lee and Kim [4]. It is also an HMM-based technique for detection of non-gestures. The purpose of the approach is similar to the one we have in the universal positive model. It is a weak model for all trained gestures in the dataset. The difference of the threshold model with our universal models is that it is not actually a trained model, but rather a “generated” model.

We have used the same training samples for the gestures in the dataset at once to train a universal positive model. However, in their proposal, they used the training data not as a whole to train their threshold model. They first trained their gesture models separately and after all of the gestures are trained, the hidden states from each gesture model, with their self-transition and observation-transition probabilities fixed, are taken and all bonded together in the threshold model. In order to provide complete transitivity, it is designed as an ergodic model. Although, it should be stated; it is possible that using an ergodic model could create some disadvantage because it does not have the temporal property that left-to-right models have. This disadvantage is critical because it is what makes this approach work on theory.

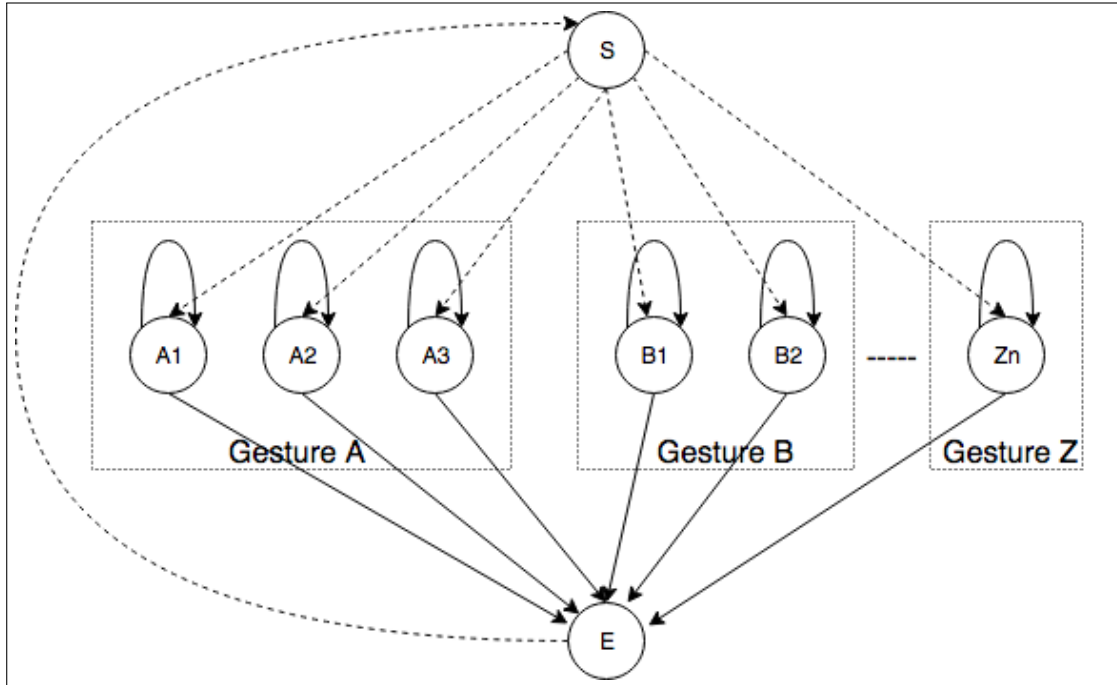


Figure 4.2: A simplified structure of the threshold model [4].

The objective is that because the threshold model will have all the states of the corresponding model, it will also be able to match the positive input gesture. However, the specifically trained model will have a better fit because it represents the temporal relation between the states better whereas the threshold model is an ergodic model.

One potential weakness of the threshold model is that naturally it has a large number of states and that causes huge performance loss in terms of processing speed. In order to overcome this weakness, they reduced the number of states based on relative entropy, which has been used as a measure of the distance between two probability distributions [4, 44]. Because we do not need to differentiate gestures from each other and we aim to capture the mistakes in each gesture, we do not have a large number of models in each run. Hence, we did not need to carry out such a reduction in the implementation of the threshold model.

It might be expected that while threshold model, having the same states as the correct model will generate a reasonable likelihood, it is also possible that when a non-gesture is not at all probable with any combination of the states in the

threshold model, it could give again a similar value like an average of the trained gestures. We suspect that problems could arise in cases where there are not many trained models, like in our case. The results provided in the study demonstrates great success in determining the non-gesture motions for the dataset used.

## 4.4 Comparison

In Tables 4.1-4.6, we observe that the universal negative model and the threshold model have similar performances overall; the universal negative model having a higher F1 score in some exercises while the threshold model having higher scores in others. The precision and recall values are also similar, so it can be concluded that these methods perform in a similar manner when creating an adaptive threshold value. However, it should be noted that the performance of the universal negative model depends on the training set. Under different conditions, reproducing the same results may not be possible.

The universal positive model performs poorly compared to the other two approaches. The precision value of the universal positive model is lower than those of the universal negative and threshold models, however, it is close to them. Nevertheless, the real difference is in the recall value. One can also observe that the number of false-negatives is much higher in the universal positive model for each gesture, hence, it leads to a significantly low recall value. This could be explained by the universal positive model having a temporal structure like the actual gesture models. One thing that gives the threshold model an advantage is its ergodic structure which makes the gesture models to have higher likelihoods for correct gestures. However, because the universal positive model also has a temporal structure (left-to-right model) it sometimes generates higher probabilities for defined gestures than the gesture models, and as a result, producing false negatives.

Table 4.1: The comparison of the models for shoulder flexion.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	51	9	58	2	0.9623	0.8500	0.9027
Universal positive	41	19	52	8	0.8367	0.6833	0.7523
Threshold	53	7	56	4	0.9298	0.8833	0.9060

Table 4.2: The comparison of the models for shoulder abduction.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	54	6	56	4	0.9310	0.9000	0.9153
Universal positive	45	15	55	5	0.9000	0.7500	0.8182
Threshold	53	7	58	2	0.9636	0.8833	0.9217

Table 4.3: The comparison of the models for external rotation.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	59	1	60	0	1.0000	0.9833	0.9916
Universal positive	39	21	49	11	0.7800	0.6500	0.7091
Threshold	53	7	57	3	0.9464	0.8833	0.9138

Table 4.4: The comparison of the models for elbow flexion and extension.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	60	0	60	0	1.0000	1.0000	1.0000
Universal positive	47	13	54	6	0.8868	0.7833	0.8319
Threshold	56	4	56	4	0.9333	0.9333	0.9333

Table 4.5: The comparison of the models for combined PNF pattern.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	49	11	52	8	0.8596	0.8167	0.8376
Universal positive	43	17	51	9	0.8269	0.7167	0.7679
Threshold	55	5	52	8	0.8730	0.9167	0.8943

Table 4.6: The overall comparison of the models.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Universal negative	273	27	286	14	0.9512	0.9100	0.9302
Universal positive	215	85	261	39	0.8465	0.7167	0.7762
Threshold	270	30	279	21	0.9278	0.9000	0.9137

## Chapter 5

# Improved Accuracy for Incorrect Exercises

We examined a variety of approaches that focus on gesture recognition. The gestures in the dataset of those works consisted of different gestures that do not usually resemble each other. We, in our dataset, also have 5 different occupational therapy exercises. However, as stated in previous chapters, our purpose is not differentiating these gestures from one another. For each gesture, we also have several types of “incorrect” gestures that are actually versions of the same gesture that is performed in an undesired way. These mistakes are determined by the therapists and according to their guidance, added to our dataset. So, the aim of this study to differentiate each gesture from their “incorrect” versions.

Because of the nature of this problem, traditional gesture recognition approaches may not perform well. This problem has some unique properties and in works that also have such characteristics, the results of traditional methods are found to be relatively poor [11]. A more tailored method needs to be engineered in order to achieve improved accuracy in differentiating these correct and incorrect gestures.

We propose two different methods as a solution: *feature thresholding* and *negative models*. The solution of the negative model is actually built upon the solution of the feature thresholding. In order to understand the significance and contribution of negative models, one needs to comprehend the reasoning behind feature thresholding.

## 5.1 Feature Thresholding

The feature set for each gesture is determined with the help of extensive testing under the supervision of occupational therapists. The features that would define the gesture in the best way possible and provide the best differentiation are chosen. However, the goal is to differentiate each gesture from its incorrect versions, not from other gestures. As a result, another issue we focus on when determining the features is the characteristics of these incorrect gestures.

For each gesture, the most common and most undesired mistakes were categorized by the occupational therapists and the joint angles that best defines these mistakes were identified. In this way, the joint angles and the critical values that should or should not be exceeded by the patient are determined for every incorrect gesture version.

The number of features that we use is restricted to be between four and six. The four features (two shoulder angles and two elbow angles) are often necessary and sufficient to define each gesture. Nevertheless, when the type of compensation moves and resulting incorrect gestures are examined, extra features need to be added.

Because CP patients often suffer from muscle stiffness, they try to compensate for this stiffness by activating (flexing, extending) other muscles in their body. Because we focus on upper extremity exercises, the compensation tendencies, and common mistakes are defined by the therapists as a result of performing each exercise with CP patients. When performing upper extremity exercises, three

main compensation techniques come up: bending the body forward or backward, bending the neck to activate the upper shoulder muscles, and bending the elbow for shoulder exercises. Hence, additional to the elbow and shoulder angles, the body and head angles are also added to the feature set of each exercise so that we can catch the incorrect gestures.

There are two types of restrictions that define a gesture as an incorrect one apart from being classified as a non-gesture:

- *Type 1*: Moving a joint that should stay fixed more than a specified value. For instance, when performing the shoulder flexion exercise; the elbow angle on the X-Z plane should stay between  $15^\circ$  and  $-15^\circ$  and the shoulder angle on the X-Z plane between  $75^\circ$  and  $105^\circ$ . These two movements, i.e., moving the shoulder forward or bending the elbow, take the tension from the shoulder muscles that should complete the exercise and make the exercise ineffective. Similar restrictions also exist for the body and head angles on different exercises.
- *Type 2*: Not reaching or exceeding a target angle. For the shoulder flexion exercise, the shoulder angle on the Y-Z plane should reach a value between  $75^\circ$  and  $105^\circ$  at its peak point, stay there for a period of time and then decrease. The real target value is  $90^\circ$  where the  $15^\circ$  tolerance value is determined considering the inaccuracy of the depth camera.

What we do for these restrictions is that we modify the input value for these features. For Type 1 restrictions, the feature value is not gradually changing during the exercise, like elbow angle on the X-Z plane for the shoulder flexion. In this case, we modify it so that it is TRUE when staying in the range specified and FALSE when it is not. For Type 2 restrictions, the feature value is naturally changing throughout the exercise, like shoulder angle on the X-Y plane for shoulder abduction. For these types of features, we record the real value of the angle until it reaches the target point. When the target point is reached we change the data to a value that is not a defined angle in our system (out of the range



between  $-360^\circ$  and  $360^\circ$ ) and fix it until the real value exceeds the range or falls below it.

## 5.2 Negative Models

Negative models are basically new models that represent the incorrect gestures. The idea is similar to the Universal Negative Model explained in Chapter 4.1. The difference is that the training set for these models consist of specific gestures together with feature thresholding. These models are trained using determined compensation mistakes as training set and using the same features as the correct gesture model. Type 1 and Type 2 restrictions are also applied during the training and recognition to increase accuracy. There are two types of negative models we propose: fault specific negative model and gesture-specific negative model.

### 5.2.1 Fault-specific Negative Model

Fault-specific gesture model is actually the basic application of the negative model concept. In this approach, a separate model is trained for each different compensation mistake. The downside of this is that the computational cost is increased as the number of the types of mistakes increases.

We compared fault specific negative model to the baseline solution that does not use any specific negative model other than the universal negative model. The only possible scenario for baseline solution to classify a compensation mistake is by classifying it as a non-gesture. Hence, the sum of false negative and true negative for each gesture equals to non-gesture count.

The results presented in Tables 5.1-5.6 show the superiority of fault-specific negative model in terms of accuracy. Because the baseline solution, which is similar to gesture recognition solutions that are used in generic exercise recognition problems, is not designed specifically to solve the problem of small mistakes;

such a difference in accuracy is expected. One can see that the baseline solution performs better in terms of recall value. This is because the baseline solution classifies most of the incorrect gestures as correct gestures and false positives are not taken into account when calculating recall. However, the objective of the fault specific approach is to reduce false positives, and in that case, the precision value is very important for comparison.

### **5.2.2 Gesture-specific Negative Model**

The difference of gesture-specific negative model is that it encapsulates all types of mistakes related to one exercise in a single model for each different exercise. The reason we applied this approach is to increase processing speed. Even though we did not have the number of mistakes to decrease the performance greatly in our tests, such a solution could be needed for different exercises or patient types.

The gesture-specific negative model is compared to the baseline solution in the same way as the fault-specific negative model (see Tables 5.7-5.12). The results show us that the gesture-specific negative model performs better than the baseline solution in terms of precision and F1 Score. The baseline solution has a better recall value overall, but as it is explained in previous parts. the recall value is not significant in this case.

### **5.2.3 Comparison**

When we compare the results of these two approaches, we see that the fault-specific negative model generates better results than the gesture-specific negative model. This is because the gesture-specific model has different gestures in its training set. The rationale for the usage of the gesture-specific approach was to reduce the computational burden. While only one extra model is calculated for the gesture-specific model, the fault-specific model requires as many models as the number of defined mistakes.

Table 5.1: The fault-specific negative model and the baseline solution for shoulder flexion.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	24	6	9	21	0.5333	0.8000	0.6400
Fault-specific model	23	7	22	8	0.7419	0.7667	0.7541

Table 5.2: The fault-specific negative model and the baseline solution for shoulder abduction.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	25	5	4	26	0.4902	0.8333	0.6173
Fault-specific model	25	5	21	9	0.7353	0.8333	0.7813

Table 5.3: The fault-specific negative model and the baseline solution for external rotation.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	29	1	4	26	0.5273	0.9667	0.6824
Fault-specific model	27	3	26	4	0.8710	0.9000	0.8852

Table 5.4: The fault-specific negative model and the baseline solution for elbow flexion and extension.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	26	4	2	28	0.4815	0.8667	0.6190
Fault-specific model	25	5	24	6	0.8065	0.8333	0.8197

Table 5.5: The fault-specific negative model and the baseline solution for combined PNF pattern.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	27	3	10	20	0.5745	0.9000	0.7013
Fault-specific model	24	6	22	8	0.7500	0.8000	0.7742

Table 5.6: The fault-specific negative model and the baseline solution for all exercises cumulatively.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	131	19	29	121	0.5198	0.8733	0.6517
Fault-specific model	124	26	115	35	0.7799	0.8267	0.8026

Table 5.7: The gesture-specific negative model and the baseline solution for shoulder flexion.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	24	6	9	21	0.5333	0.8000	0.6400
Gesture-specific model	22	8	19	11	0.6667	0.7333	0.6984

Table 5.8: The gesture-specific negative model and the baseline solution for shoulder abduction.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	25	5	4	26	0.4902	0.8333	0.6173
Gesture-specific model	23	7	18	12	0.6571	0.7667	0.7077

Table 5.9: The gesture-specific negative model and the baseline solution for external rotation.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	29	1	4	26	0.5273	0.9667	0.6824
Gesture-specific model	25	5	20	10	0.7143	0.8333	0.7692

Table 5.10: The gesture-specific negative model and the baseline solution for elbow flexion and extension.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	26	4	2	28	0.4815	0.8667	0.6190
Gesture-specific model	21	9	19	11	0.6563	0.7000	0.6774

Table 5.11: The gesture-specific negative model and the baseline solution for combined PNF pattern.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	27	3	10	20	0.5745	0.9000	0.7013
Gesture-specific model	22	8	23	7	0.7586	0.7333	0.7458

Table 5.12: The gesture-specific negative model and the baseline solution for all exercises cumulatively.

	<b>TP</b>	<b>FN</b>	<b>TN</b>	<b>FP</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>
Baseline solution	131	19	29	121	0.5198	0.8733	0.6517
Gesture-specific model	113	37	99	51	0.6890	0.7533	0.7197

# Chapter 6

## Evaluation and Results

We proposed improvements to conventional gesture recognition methods in order to provide a better solution to the problem at hand. In previous chapters, we compared our proposed solutions to state-of-the-art approaches and the baseline solutions and presented the resulting data. In this chapter, we focus on the overall results of our proposed solution and present the results of using our solution in occupational therapy.

The results of the test sessions are provided in Chapters 4 and 5. We describe the method we used when conducting the tests here. During our study, constant testing with occupational therapists and CP patients took place. Our target users were hemiplegic CP patients that are classified in levels 1 or 2 of GMFCS and in levels 2 or 3 of MACS standards. Six children with CP between the ages of 7 and 12 were chosen for performing the exercises.

A total of six detailed testing sessions were completed in a span of 24 weeks. Each session lasted 45-60 minutes. The results presented in Chapters 4 and 5 belong to the last session. Previous sessions are performed for different reasons: restricting the number of features, selecting features, defining compensation mistakes, tracking the children's progress, and so on.

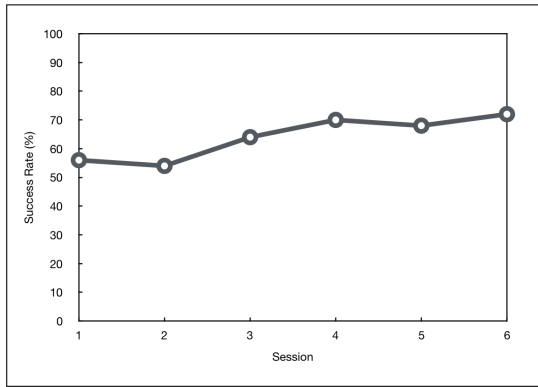
By means of collecting the data for all six sessions, we are able to observe the

patients' performance and evaluate the overall benefits of our solution. Nevertheless, it should be noted that the results obtained here do not prove that the improvements to their performance is solely the result of using our solution. During this phase, these children were continuing their conventional rehabilitation programs and were also having exercise sections in related facilities. Restricting their rehabilitation program to our solution is not possible and creating control groups having similar levels of complications is demanding medically and also requires special permissions.

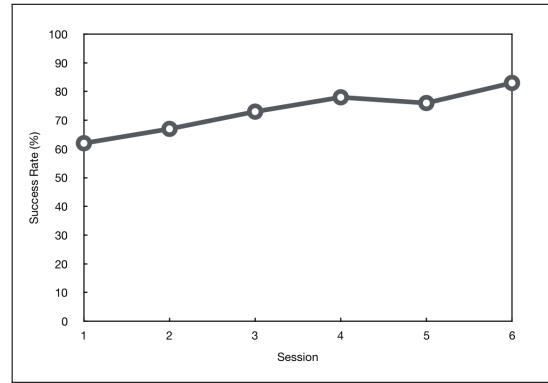
We use the success rate to evaluate children's progress during this study. It is simply the ratio of correctly performed exercises to all gestures performed by the children. A parallel study conducted by occupational therapist used a different method to measure the progress of children. Dynamic Occupational Therapy Cognitive Assessment for Children (DOTCA-Ch) is used to assess children. All data were collected strictly anonymously by an experienced therapist who was blind to the treatment protocol. DOTCA-Ch also evaluates the child's cognitive state.

The pre-intervention scores of DOTCA-Ch were  $3.81 \pm 2.26$  in orientation,  $5.27 \pm 2.09$  in motor control and  $15.72 \pm 8.51$  in visuomotor construction. After the last session is completed, the orientation score was improved to  $5.09 \pm 2.15$ , motor control was at  $6.09 \pm 1.77$  and visuomotor construction was  $18.54 \pm 7.77$ . These measures show a significant statistical difference in performance. It should be noted that the cognitive abilities are also taken into account.

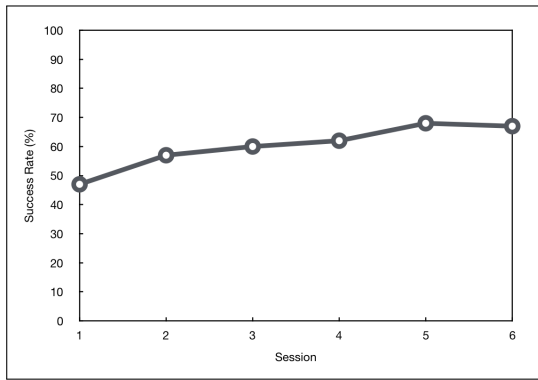
Figure 6.1 shows the improvement in children's success rates when performing the determined five exercises. The presented data are the cumulative result of all children. A gradual increase is observed for all five exercises, PNF pattern showing the least improvement.



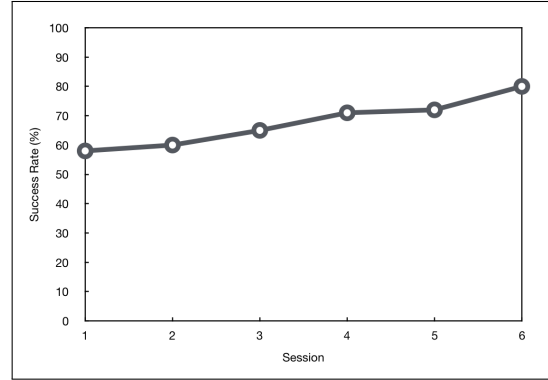
(a)



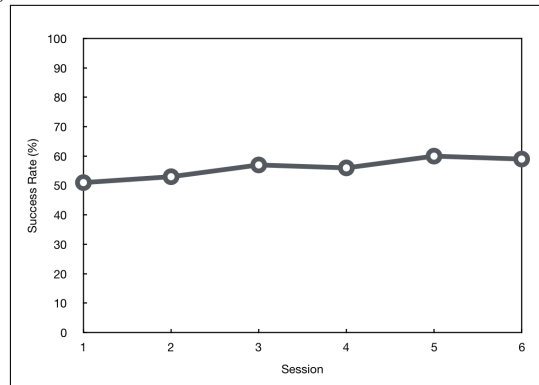
(b)



(c)



(d)



(e)

Figure 6.1: The success rate graph for (a) shoulder flexion, (b) shoulder abduction, (c) external rotation, (d) elbow flexion and extension, and (e) combined PNF pattern.

# Chapter 7

## Conclusions and Future Research Directions

We propose a new approach that makes it possible to use gesture recognition for occupational therapy exercises with children with cerebral palsy. In order to differentiate gestures that are not defined exercises, which is an important problem in our case considering the cognitive impairments of the children, we proposed an alternative method called universal negative model and universal positive model. The purpose of these methods is to generate an adaptive threshold model. We were able to get similar results compared to a successful method in the literature. We also proposed various solutions for capturing the exercise mistakes done by the patients in order to compensate for their lack of muscle control and muscle strength. These incorrect exercises generally resemble the original exercise and thus classified as a correct exercise by traditional gesture recognition algorithms. With the help of our new approach, it is possible to get reasonable results compared to the conventional approach.

Because this is not a problem that is dealt with before by other studies it is not possible to make a direct comparison. Other approaches focus on other aspects of gesture recognition whereas we focus on capturing the compensation mistakes. However, the effects of our approach on children's motor control and orientation



progress are examined and a significant improvement is observed.

A very basic method to separate the time frames of each gesture from one another is implemented. We used the starting and ending poses of each gesture for this purpose. As future work, a sliding-window based method could be used for better results. Recent developments in deep learning methods show that these approaches could be utilized. Adapting the solutions we proposed to deep neural network approaches could be a good research direction in the future.

# Bibliography

- [1] L. Xia, C.-C. Chen, and J. Aggarwal, “View invariant human action recognition using histograms of 3D joints,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 20–27, 2012.
- [2] M. Z. Uddin, “Human activity recognition using body joint-angle features and hidden Markov model,” *ETRI Journal*, vol. 33, no. 4, pp. 569–579, 2011.
- [3] D. Webster and O. Celik, “Experimental evaluation of Microsoft Kinect’s accuracy and capture rate for stroke rehabilitation applications,” in *Proceedings of the IEEE Haptics Symposium*, HAPTICS ’14, pp. 455–460, 2014.
- [4] H.-K. L. H.-K. Lee and J. Kim, “An HMM-based threshold model approach for gesture recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961–973, 1999.
- [5] D. Webster and O. Celik, “Systematic review of Kinect applications in elderly care and stroke rehabilitation,” *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 108, 2014.
- [6] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.
- [7] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, “A survey on human motion analysis from depth data,” in *Time-of-Flight and Depth Imaging: Sensors, Algorithms, and Applications, Lecture Notes in Computer Science*, vol. 8200, pp. 149–187, Springer, 2013.

- [8] A. Delia Calin, “Gesture recognition on Kinect time series data using Dynamic Time Warping and Hidden Markov Models,” in *Proceedings of the 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pp. 264–271, 2016.
- [9] K. K. Biswas and S. K. Basu, “Gesture recognition using Microsoft Kinect,” in *Proceedings of the 5th International Conference on Automation, Robotics and Applications*, vol. 2, pp. 100–103, IEEE, 2011.
- [10] N. H. Dardas and N. D. Georganas, “Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.
- [11] V. Bloom, D. Makris, and V. Argyriou, “G3D: A gaming action dataset and real time action recognition evaluation framework,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7–12, 2012.
- [12] S. Sempena, N. U. Maulidevi, and P. R. Aryan, “Human action recognition using Dynamic Time Warping,” in *Proceedings of the Electrical Engineering and Informatics, ICEEI ’11*, pp. 1–5, 2011.
- [13] M. Bicego, U. Castellani, and V. Murino, “Using hidden Markov models and wavelets for face recognition,” in *Proceedings of the 12th International Conference on Image Analysis and Processing*, pp. 52–56, IEEE Computer Society, 2003.
- [14] S. Ong and S. Ranganath, “Automatic sign language analysis: A survey and the future beyond lexical meaning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, 2005.
- [15] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, “Invariant features for 3-D gesture recognition,” in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 157–162, 1996.

- [16] B. Schuster-Böckler and A. Bateman, “An introduction to hidden Markov models,” *Current Protocols in Bioinformatics*, vol. 18, no. 1, pp. Appendix 3A, 9 pages, 2007.
- [17] L. R. Rabiner, “A Tutorial on Hidden Markov Models and selected applications in speech recognition,” 1989.
- [18] H. Haberdar, “Saklı Markov Model Kullanılarak Görüntüden Gerçek Zamanlı Türk İşsaret Dili Tanıma Sistemi,” Master’s thesis, Department of Computer Science, Yıldız Technical University, 2005.
- [19] J. G. D. Forney, “The Viterbi Algorithm,” *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973.
- [20] N. Liu and B. C. Lovell, “Gesture classification using hidden Markov models and Viterbi path counting,” in *Proceedings of the VIIth Digital Image Computing: Techniques and Applications*, pp. 10–12, 2003.
- [21] Z. Yang, Y. Li, W. Chen, and Y. Zheng, “Dynamic hand gesture recognition using hidden Markov models,” in *Proceedings of the 7th International Conference on Computer Science & Education, ICCSE ’12*, pp. 360–365, 2012.
- [22] M. Z. Uddin, N. Thang, and T.-S. Kim, “Human activity recognition via 3-D joint angle features and hidden Markov models,” in *Proceedings of the International Conference on Image Processing, ICIP ’10*, pp. 713–716, 2010.
- [23] A. O. T. Association, “Occupational therapy practice framework : Domain and process,” *American Journal of Occupational Therapy*, vol. 56, pp. 609–639, 2002.
- [24] H. M. H. Pendleton and W. Schultz-Krohn, *Pedretti’s Occupational Therapy - E-Book: Practice Skills for Physical Dysfunction*. Factsbook, Elsevier Health Sciences, 2013.
- [25] W. Y. W. Yu, R. Dubey, and N. Pernaletе, “Robotic therapy for persons with disabilities using Hidden Markov Model based skill learning,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2 of *ICRA ’04*, pp. 2074–2079, 2004.

- [26] Y.-J. Chang, W.-Y. Han, and Y.-C. Tsai, “A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy,” *Research in Developmental Disabilities*, vol. 34, pp. 3654–3659, nov 2013.
- [27] 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*, vol. 32, no. 6, pp. 2566–2570, 2011.
- [28] M. Pedraza-Hueso, S. Martín-Calzón, F. J. Díaz-Pernas, and M. Martínez-Zarzuela, “Rehabilitation using Kinect-based games and virtual reality,” *Procedia Computer Science*, vol. 75, no. Vare, pp. 161–168, 2015.
- [29] M. Bax, M. Goldstein, P. Rosenbaum, A. Leviton, N. Paneth, B. Dan, B. Jacobsson, and D. Damiano, “Proposed definition and classification of cerebral palsy,” *Developmental Medicine & Child Neurology*, vol. 47, pp. 571 – 576, 2005.
- [30] L. Reid, S. E Rose, and R. Boyd, “Rehabilitation and neuroplasticity in children with unilateral cerebral palsy,” *Nature Reviews, Neurology*, vol. 11, 2015.
- [31] M. Mutsaerts, B. Steenbergen, and H. Bekkering, “Anticipatory planning of movement sequences in hemiparetic cerebral palsy,” *Motor Control*, vol. 9, pp. 439–58, 2005.
- [32] K. Berger, S. Meister, R. Nair, and D. Kondermann, “A state of the art report on research in multiple RGB-D sensor setups,” in *Time-of-Flight and Depth Imaging: Sensors, Algorithms, and Applications*, vol. 8200, pp. 257–272, Springer, 2013.
- [33] A. Corti, S. Giancola, G. Mainetti, and R. Sala, “A metrological characterization of the Kinect V2 time-of-flight camera,” in *Robotics and Autonomous Systems*, vol. 75, Part B, pp. 584–594, Elsevier, 2016.
- [34] R. A. Clark, Y. H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant, “Validity of the Microsoft Kinect for assessment of postural control,” *Gait and Posture*, vol. 36, no. 3, pp. 372–377, 2012.

- [35] B. Bonnechère, B. Jansen, P. Salvia, H. Bouzahouene, L. Omelina, F. Moiseev, V. Sholukha, J. Cornelis, M. Rooze, and S. Van Sint Jan, “Validity and reliability of the Kinect within functional assessment activities: Comparison with standard stereophotogrammetry,” *Gait and Posture*, vol. 39, no. 1, pp. 593–598, 2014.
- [36] 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 and Posture*, vol. 39, no. 4, pp. 1062–1068, 2014.
- [37] R. T. Collins, R. Gross, and J. Shi, “Silhouette-based human identification from body shape and gait,” in *Proceedings of the 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR ’02*, pp. 366–371, 2002.
- [38] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3D points,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–14, 2010.
- [39] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.
- [40] T. P. Mann, “Numerically stable hidden Markov model implementation,” *An HMM scaling tutorial*, pp. 1–8, 2006.
- [41] N. D. Binh, E. Shuichi, and T. Ejima, “Hand gesture recognition using a real time tracking methods and pseudo hidden Markov model,” *International Conference on Graphics, Vision and Image Processing*, pp. 362–368, 2005.
- [42] X. Zabulis, H. Baltzakis, and A. Argyros, “Vision-based hand gesture recognition for human-computer interaction,” in *The Universal Access Handbook*, pp. 341 – 343, CRC Press, 2009.

- [43] Y. Dennemont, G. Bouyer, S. Otmane, and M. Malle, “A discrete Hidden Markov models recognition module for temporal series: Application to real-time 3D hand gestures,” in *Proceedings of the 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2012.
- [44] T. M. Cover and J. A. Thomas, “Entropy, relative entropy, and mutual information,” in *Elements of Information Theory*, pp. 12–49, John Wiley & Sons, 2001.