EVALUATING PHYSICAL REHABILITATION WITH A KINECT BASED LOW COST SYSTEM

By

USMAN ZAFAR

Bachelor of Science in Computer Science

National University of Computer & Emerging Sciences

Lahore, Pakistan

2010

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2013

EVALUATING PHYSICAL REHABILITATION WITH A KINECT BASED LOW COST SYSTEM

Thesis Approved:

Dr. Christopher Crick

Thesis Adviser

Dr. Weihua Sheng

Dr. Blayne E. Mayfield

Name: USMAN ZAFAR

Date of Degree: JULY, 2013

Title of Study: EVALUATING PHYSICAL REHABILITATION WITH A KINECT BASED LOW COST SYSTEM

Major Field: COMPUTER SCIENCE

Abstract:

In this thesis a low cost low power embedded system is presented which enables the patients to evaluate the performance of their physical rehabilitation. The system uses the Asus Xtion as a motion sensor which is connected to a portable Beagleboard XM (an open source single board small computer). In order to enable the skeleton tracking functionality on the BeagleBoard XM, the Beckon SDK (provided by Omek Interactive) is used. The physical rehabilitation system uses the Beckon SDK to retrieve the joint positions from each frame and compute elbow and shoulder angles. Initially the instructor records the desired exercise and then the patient performs the same exercise. To identify the start and end of the exercise, an additional gesture has been introduced. Next, to remove redundant and noisy data from sequences, curve extraction and median filter algorithms are used. Then a comparison algorithm known as DTW (Dynamic Time Warping) is used to compare the angular sequences and compute a total score which represents a quantitative evaluation of the exercise. For visual feedback a QT interface is used. Instructions are provided on the screen to tell both the instructor and patient about what to do next. Real time angular data graphs as well as frames per second (FPS) are also displayed on screen for the user’s benefit. Experiments were conducted to compare the system performance between low cost embedded systems and high end desktop machines. The skeleton tracking performance is also evaluated using a VICON motion capture system.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION

II. REVIEW OF LITERATURE

III. SYSTEM DESIGN

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This chapter briefly describes the motivation behind the physical rehabilitation system on which this thesis is based, as well as an overview describing how the system works and what the requirements of the system are.

1.1 Motivation

Physical exercise is a part of human life, as well as an important component of treatment for many diseases. In this modern world where technology is growing increasingly more sophisticated, people are trying to find different ways to substitute machines for humans in different aspects of life. In consequence, many people have less access to physical work. This not only affects their health but also reduces their overall efficiency and productivity. In recent years several games [12] [13] [5] [6], applications [3] [14] and TV programs have appeared to help humans exercise. However, in the realm of physical therapy, every patient has different exercise needs and must design their regimen in consultation with therapist. The games and applications which are available nowadays have pre-configured exercises from which a patient must choose from [3]. Furthermore the systems on which these applications or games execute are usually expensive. The motivation of this thesis is to provide patients with a low cost system which they can use to evaluate their exercise in the comfort of their home.

1.2 Overview of the physical rehabilitation system

The physical rehabilitation system demonstrated in this thesis runs on a portable low cost hardware platform known as Beagleboard XM [17]. It uses the Beckon SDK [11] to get the human body's joint locations. Then it calculates the elbow and shoulder angles during instructor and patient movements. After that it uses different algorithms to compare the angle sequences recorded by instructor and patient. The final score provided to the user depicts how accurately he/she has performed as compared to the instructor movements. For example, once we start the system, it first asks the instructor or the physical therapist to record the desired movement. After that the user can take the system to his/her home and perform the same exercise to get the feedback. The user can use the system several times to get the feedback. The system is easy to carry which gives user an opportunity to place the system anywhere in the home. It is also customizable which gives physical therapist an opportunity to record the desired movement. Furthermore, it provides valuable feedbacks in the form of scores and graphs. It requires an Asus Xtion Pro [21] to receive depth based images, a SD Card to boot the system operating system and the rehabilitation application, and a battery to power the Beagleboard XM. For display an LCD touch screen device can be used.

CHAPTER II

REVIEW OF LITERATURE

In this chapter, work which is closely related to this thesis is briefly described. The issues which most of the researchers have ignored in their research work are specifically addressed in this thesis. Many researchers are using video games to motivate patients for physical rehabilitation. Some researchers have introduced performance evaluation in applications for physical rehabilitation in which the physical therapists either have to provide the data manually or the application gives them the ability to record their data automatically.

2.1 Physical rehabilitation using games

Pastor et al. [12] have developed computer games which used the Kinect [20] to provide upper elbow rehabilitation for stroke patients. Users use Kinect to control a cursor and then use that cursor to click on the images that appear on the grid on screen. The location of the image and the resolution of the grid define the exercise and its difficulty level. Lange et al. [13] developed an interactive game based rehabilitation system using the Kinect which displays gems on the screen and then the user has to click on the gems if they are glowing. The sequence of clicking on glowing gems defines the rehabilitation movement for the patient. Guerts et al. [5] presented four minigames designed and developed especially for people with motor disabilities. Physical therapists took part in the design phase of the games so that the games can make the patients do specific movements which physical therapists want them to do. Ganesan et al. [6] presented a project that aimed to find the factors that play an important role in motivating older adults to maintain a physical exercise routine, a habit recommended by doctors but difficult to sustain. Their initial data gathering includes an interview with an expert in aging and physical therapy, and a focus group with older adults on the topics of exercise and technology. Based on these data, an early prototype game was implemented using the Kinect that was aimed to encourage older adults to exercise. They reported that the Kinect application was tested for basic usability and found to be promising. Chang et al. [7] presented a performance based comparison on motion tracking between the low-cost Kinect and the high fidelity OptiTrack optical system [19]. Data was collected on six upper elbow motor tasks that have been incorporated into game-based rehabilitation applications. The experiment results showed that Kinect can achieve motion tracking performance competitive with OptiTrack and provide pervasive accessibility that enables patients to take rehabilitation treatment in both clinic and home environment.

In all the above studies, researchers have used games to provide physical rehabilitation for the patients. The advantage of using games is to motivate patients to do physical rehabilitation. However games have disadvantages such as high console hardware cost and a lack of valuable feedback provided to the patients beyond winning or losing. Most importantly, console games must be programmed in advance, incurring significant development costs and preventing therapy regimens specifically customized to an individual patient’s needs. This thesis specifically addresses these issues to provide patients with a more flexible, efficient and low cost physical rehabilitation system which is easy to carry and easy to use. This new methodology not only lets both physical therapists and patients record their movements but also provides valuable feedbacks to the user through interactive interfaces.

2.2 Physical rehabilitation with performance evaluation

Bajcsy et al. [2] presented a performance based evaluation between the Kinect and a motion capture system. They did a quantitative assessment of subject movements through reachable surfaces by comparing it with the motion capture system. The results showed that the Kinect based measurement is sufficiently accurate and robust for this type of evaluation. These results also motivated them to evaluate patients in clinical settings. Huang et al. [3] presented the Kinerehab application, which uses the Microsoft's Kinect motion sensor with an integrated database, video instruction, and voice reminders to form an intelligent rehabilitation system. The system automatically detects the student's joint position, and uses the data to determine whether the student's movements have reached rehabilitation standards or not. Using their application, the instructor can create target exercise for patients by selecting movements from a predefined set of movements.

In the work of Huang et al., researchers have created an application which requires the presence of a physical therapist to evaluate the patient rehabilitation process. The physical therapist either has to fill the questionnaire form to provide real data or has to customize exercise from the given predefined set of limited moves. All of these methods restrict the physical therapist's ability to record a complex series of movements which can further be used to compare with patients movements. This thesis addresses this issue too. In this work physical therapists are able to record a series of movements once and this data then can be used to compare with patient's movements.

Gama et al. [8] did an analysis of the use of the Kinect sensor as an interaction support tool for rehabilitation systems. The Kinect sensor gives three-dimensional information about the user body, enabling the extraction of skeleton and joint positions; however it does not provide body specific movements. In this way, the correct description of a rehabilitation movement (shoulder abduction, for instance) was implemented in a system prototype. Their project was undertaken to recognize the shoulder abduction movement where shoulder and elbow angles were computed and evaluated to analyze the correctness of the movement. A scoring mechanism was also developed in order to measure the patient performance. Su et al. [14] developed a home based rehabilitation system which runs on a desktop machine. They used the Kinect for Windows SDK to get the three dimensional X, Y and Z location of the joints and then compared the trajectories of each of X, Y and Z coordinates for each of the joints with trajectories previously recorded under the supervision of a professional. For comparison they used the Dynamic Time Warping (DTW) algorithm [15].

Su et al.'s work automated the recording of physical therapist data in their physical rehabilitation system, in a fashion similar to our approach. However, their application uses high end, expensive, non-portable desktop computers. This thesis on the other hand uses low cost hardware and can be set up for use anywhere. Furthermore they record the coordinate trajectories of both the instructor and the patient for comparison using the Dynamic Time Warping algorithm. In this thesis, on the other hand angle sequences are being compared which saves a great deal of computational power, without comprising the accuracy of the comparison. Additionally, in the above work researchers have not emphasized removing noisy data. In this case, our application uses filtering and curve extraction algorithms to remove noisy and redundant data. This makes the final score more reliable.

2.3 Summary

Based on the research work presented above, all of the current physical rehabilitation systems are targeted for high end machines like desktop computers or gaming consoles. Some systems are based on games to motivate patients to do exercise. Some require the presence of a physical therapist at the time of patient evaluation whereas some give the physical therapist functionality to record data beforehand. This thesis addresses these issues found in the above work. First of all this system uses low cost hardware which minimizes the overall cost of the system and provides portability and flexibility. Secondly this work provides the physical therapist the ability to record desired sequences of movements so that the system can use this data to evaluate patient exercises. The results are compared with a desktop machine and the VICON motion capturing system [18] to ensure the reliability and efficiency of the system.

CHAPTER III

SYSTEM DESIGN

In this chapter the proposed system hardware and software platforms are described in detail. The significant features of the system are as follows. 1) The cost of the system is low. 2) It is portable. 3) The system supports customized training which means the instructor can record an desired exercise based on an individual patient's needs and 4) It provides quantitative evaluation of the exercise with visual feedback. Instead of the Kinect, we use the similar Asus Xtion because the Xtion does not need a separate power source. Furthermore to save power and make the system more portable while maximizing the computational power, the Beagleboard XM is used. This board only needs 5V of input power which can be easily provided by a battery. On the software side, the physical rehabilitation application which is presented in this thesis uses the Beckon SDK and QT framework [22].

3.1 System design overview

The proposed system is comprised of the Asus Xtion to capture depth-based images, the Beagleboard XM for all the computational work, an SD card containing the Angstrom OS, Beckon SDK and the rehabilitation application, and a battery to power the Beagleboard XM. For display a touch screen LCD is used. Figure 1 shows the block diagram of the system.

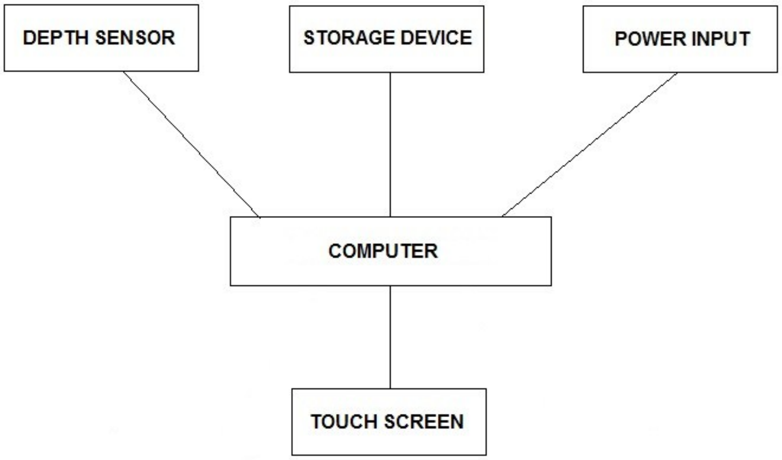


Figure 1 - Block system diagram

3.2 Hardware platform

The hardware part of the system consists of the following components.

3.2.1 Asus Xtion Pro

Figure 2 - Asux Xtion Pro [25]

The Asus Xtion Pro as shown in Figure 2 is a motion sensing device which provides real time motion sensing that captures the body movements of users. This device is used to capture the skeleton data of a human. The device has a range between 0.8m and 3.5m. The frame rate of VGA depth based images with the resolution of 640x480 is 30fps whereas the frame rate of QVGA depth based images with the resolution of 320x240 is 60fps. The device has a USB 2.0 interface which can connect to the Beagleboard XM.

3.2.2 Beagleboard XM

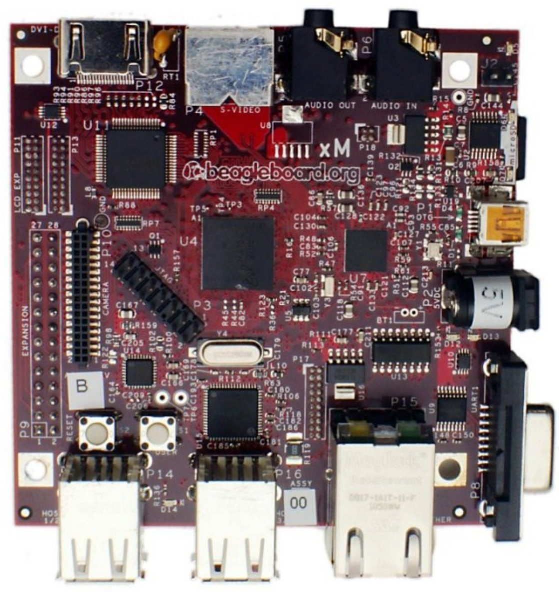


Figure 3 - Beagleboard XM [24]

The motivation to use the Beagleboard XM is that it is a low cost low power single board portable computer. As shown in Figure 3, the board has an ARM Cortex A8 processor and memory with 512MB low power DDR RAM. It can be powered with either a 5V power input or a high speed USB 2.0 OTG port. It has four on-board high speed USB ports along with a 10/100 Ethernet port. Digital monitors or TVs can be connected to the board using DVI-D port. Small LCDs can also be connected using LCD expansion headers. The board has a microSD card slot which we can use to boot the board with the operating system mounted in it. It also supports stereo audio in/out.

3.2.3 Battery

In order to make the system portable, we provide the system with a 9.6V battery and a 9.6V to 5V converter to provide 5V input to the Beagleboard XM. Fully charged, the battery can power the board for more than 3 hours.

3.2.4 4GB+ microSD Card

In order to boot the Beagleboard with the Angstrom OS, a microSD card of size 4GB+ is needed. The OS can be mounted on the microSD card and then the board can be booted by putting the SD card into its microSD card slot. The recommended producers for microSD card are Transcend and Amazon. Example of an SD card is shown in Figure 4.

Figure 4 - SD card [26]

3.2.5 LCD Touchscreen/Monitor/TV

Digital monitors and TVs can be connected to the board using its DVI-D port in order to display the output. Small size LCD touch screen devices can also be connected to the Beagleboard XM using the LCD expansion headers mounted on the back side of the board.

These hardware components are assembled using two small thin plexiglass sheets placed one over another. These are connected and spaced using long screws in the corners. After that the Beagleboard XM and Asus Xtion are attached to the upper plexiglass sheet on opposite sides. The battery and converter are mounted on the lower glass plexisheet. Output devices like touch screen LCD can be attached to the Beagleboard XM easily. The connection between these hardware components is shown in Figure 5.

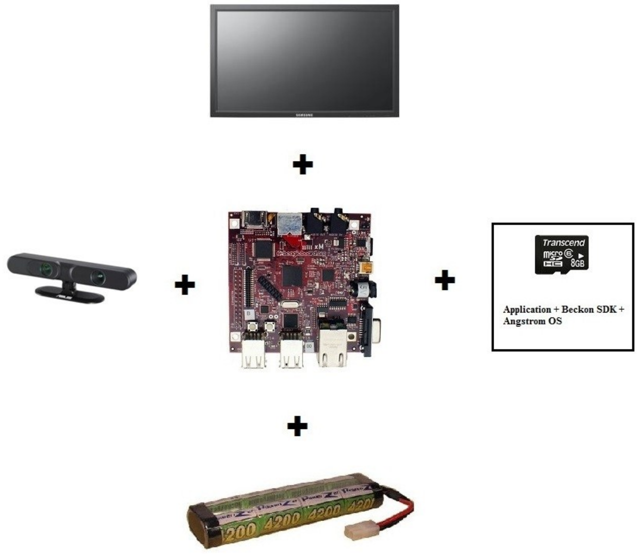


Figure 5 - System hardware components

Figure 6 shows the actual hardware setup.

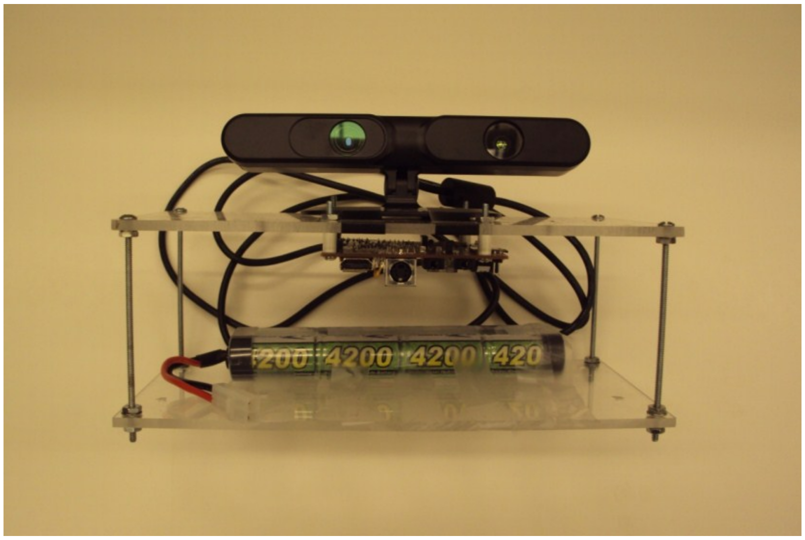


Figure 6 - The overall look of the actual complete system

3.3 Software platform

The software part of the system consists of the Angstrom OS, the Beckon SDK, the QT framework and the physical rehabilitation application. The Beckon SDK, which consists of the Beckon Engine, sensor drivers, tracking and gesture recognition algorithms, is released by Omek Interactive [11] to provide skeleton tracking technology in low cost systems like the Beagleboard XM. The main part of the Beckon SDK is installed on a Linux PC, while applications for the Beagleboard XM are cross-compiled. The Beckon engine is installed on Beagleboard XM along with sensor drivers, tracking algorithms and the QT framework. The physical rehabilitation application uses the Beckon tracking algorithms to retrieve the human skeleton and the location of the joints using the Asus Xtion motion sensor. The QT framework is used for user interfaces. For optimal performance, the application requires the user to stand in front of Asus Xtion within the range of 1.5 to 3.5 meters. Figure 7 shows the software architecture of the system.

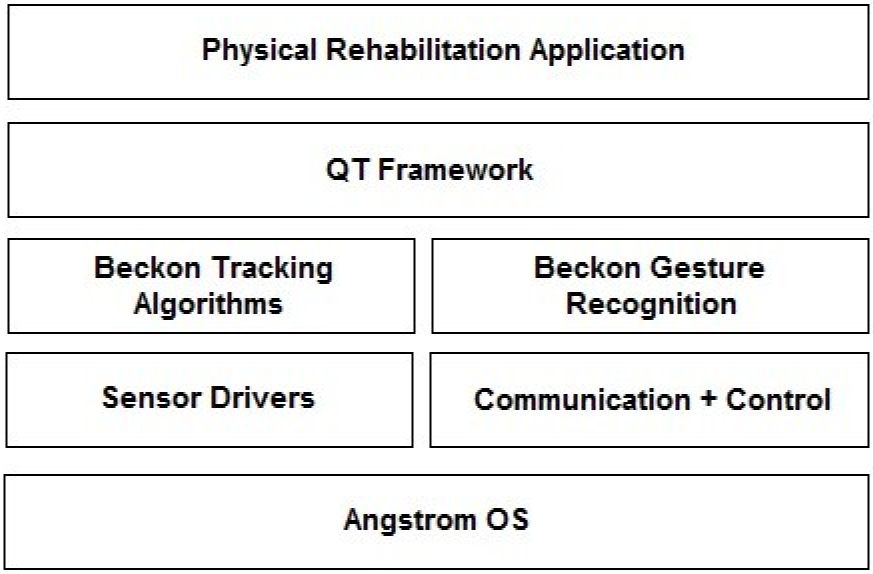


Figure 7 - Software architecture of the system

3.3.1 Software installation

The first part in installation is to mount the SD card with the Angstrom OS and the Beckon SDK. This can be accomplished by visiting the Omek Interative website and downloading their Beckon SDK SD card image for the Beagleboard XM. After downloading the image, it can be mounted on the SD card using the commands mentioned in their user manual [11]. In order to execute those commands a desktop machine with a Linux operating system like Ubuntu is needed. Those commands can be executed in the terminal of the Linux machine once you attach the SD card with the computer using any USB card reader. After mounting the image on to the SD card, the card can then be put into the slot of the Beagleboard XM and now the Beagleboard is ready to boot up.

3.3.2 Beckon SDK

To accomplish joint tracking functionalities on Beagleboard XM using Asus Xtion, the system uses Beckon SDK which consists of Beckon Engine, Sensor drivers and Beckon tracking algorithms provided by Omek Interactive. It runs on Angstrom OS on Beagleboard XM. Beckon SDK is based on two parts. The first one is for Beagleboard XM which consists of the components described above. The second part of the Beckon SDK contains a tool which is installed on the Linux PC for cross-compiling. It enables the development of tracking based applications which can be executed on the Beagleboard XM. Figure 8 shows a brief architecture of the Beckon SDK.

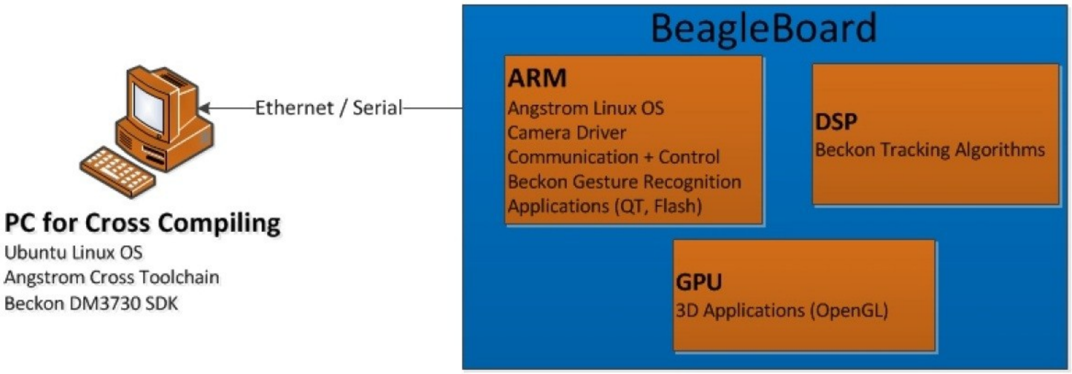


Figure 8 - Cross compilation of application written for Omek SDK [11]

The human skeleton tracking functionality in the Beckon SDK provides 23 joints for full body tracking and 17 joints for upper body tracking. Figure 9 shows the joints in the human body which can be tracked by the Beckon SDK.

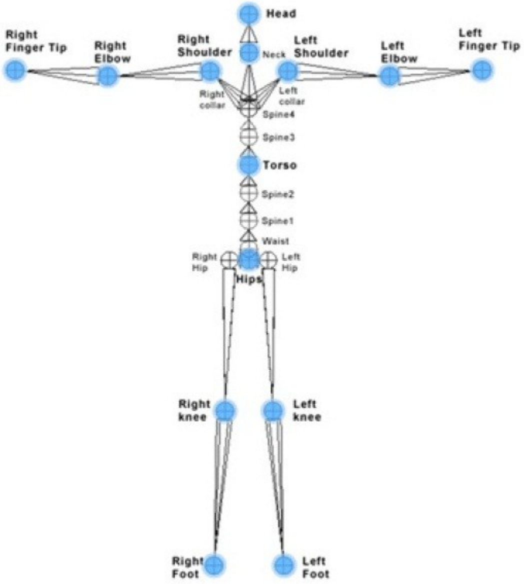
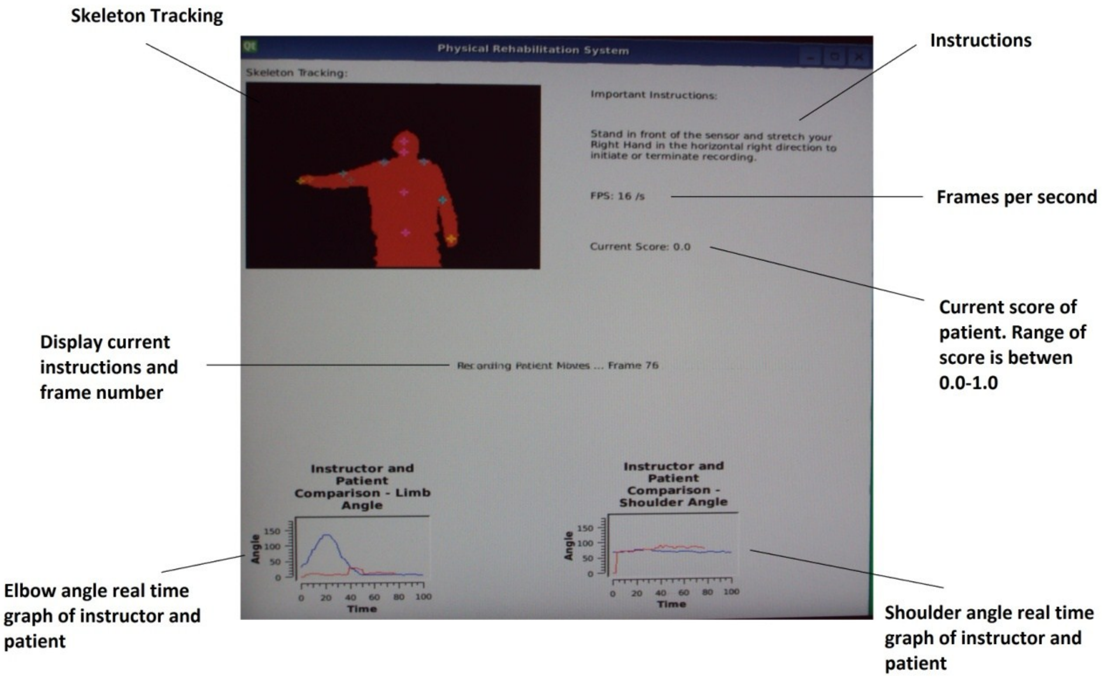


Figure 9 - Human skeleton representing joint locations [11]

3.3.3 User Interface

For visual feedback to the patients an interface created in QT is also provided by the application. QT is a cross-platform application framework that is widely used for developing software with graphical user interfaces.

In the physical rehabilitation system the main interface is created using QT. The application contains one image window displaying skeleton tracking in real time. Instructions are provided in a separate panel guiding the instructor and the patient what to do next. This thesis focuses on elbow and shoulder angles. To give more valuable feedback, separate real time graphs are also added into the interface which represent the current elbow and shoulder angle data recorded by the instructor and the patient. To enable real time graph functionality into the application, we installed the QWT library over the QT framework. The detailed installation of this library is mentioned in Appendix A. Once the patient is finished with recording, a score will be displayed based on the scoring methodology described in the next chapter. If the patient receives a high score then it means he/she performed more accurately according to the instructor’s standards. Likewise, a lower score indicates that the patient didn’t perform well. The application interface is shown in Figure 10.



Important Instructions:

Stand in front of the sensor and stretch your Right Hand in the horizontal right direction to initiate or terminate recording.

FPS: 16 /s

Current Score: 0.0

Recording Patient Moves ... Frame 76

Skeleton Tracking

Instructions

Frames per second

Current score of patient. Range of score is betwen 0.0-1.0

Display current instructions and frame number

Elbow angle real time graph of instructor and patient

Shoulder angle real time graph of instructor and patient

Figure 10 - Application user interface

CHAPTER IV

GESTURE SCORING METHODOLOGY

In this chapter the system tracking and scoring methodology, as well as the algorithms used, are described in detail. To give a patient valuable feedback about his/her performance, a scoring mechanism is defined which evaluates the patient's performance for a particular exercise. This thesis primarily focuses on two angles, the elbow and shoulder of the right arm. Other angles can also be included to enable scoring of complex exercises.

4.1 Overview

After starting the system, the user should stand in front of the Asus Xtion. For best results, the user should stand within the range of 1.5 to 3.5 meters of the depth sensor. Once the user is in range and the system is able to track his/her skeleton, it will start retrieving his/her joint locations. Once the joint locations are retrieved from a frame, vectors are created to calculate the angles between different body joints. Right now this thesis primarily focuses on the elbow and shoulder angles of the right arm. Other angles can also be calculated based on needs. To identify the start and end of the sample angular data sequence, an initial and ending gesture has been defined, with one's right arm held horizontally and straight. This will let the system know when to record and when to stop.

Furthermore to ensure the correctness of data in an angular sequence, curve extraction methodology has been introduced to discard redundant data. Initially the instructor will record his/her exercise and then one or more patients will record their exercises. Once the patient has finished, the instructor’s angular data sequences will be compared with the patient angular data sequences. The final score, which is used in order to judge whether the patient has performed the particular exercise appropriately, will be calculated using the DTW (Dynamic Time Warping) comparison algorithm. The algorithm takes two sequences of data and then computes the distance between them. Using this methodology a score is generated at the end of each exercise to let the patient know whether he/she has performed correctly or poorly. If the score is high it means the patient performed well, whereas if the score is lower than the patient did not meet the instructor’s requirements and the exercise should be repeated.

4.2 Angle calculation

The angles are calculated after the application successfully retrieves joint positions from each frame. For example as shown in Figure 11, vectors are computed from the right hand finger tip, elbow and shoulder joint positions. Vector v1 is created between the elbow and shoulder joint, while vector v2 represents the elbow and finger tip joint.

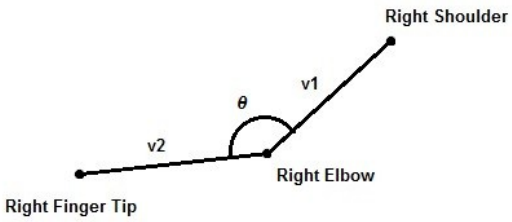


Figure 11 - Vector v1 and v2 computed from three joints

Using the same method, two more vectors are created using the elbow, shoulder and torso joints. The magnitude of the vectors can be calculated using the following formula.

$$|v_i| = \sqrt{{v_{ix}}^2 + {v_{iy}}^2 + {v_{iz}}^2} \qquad\qquad 1 \le i \le 2$$

Once we have the magnitude of both vectors, we can get the unit vectors using

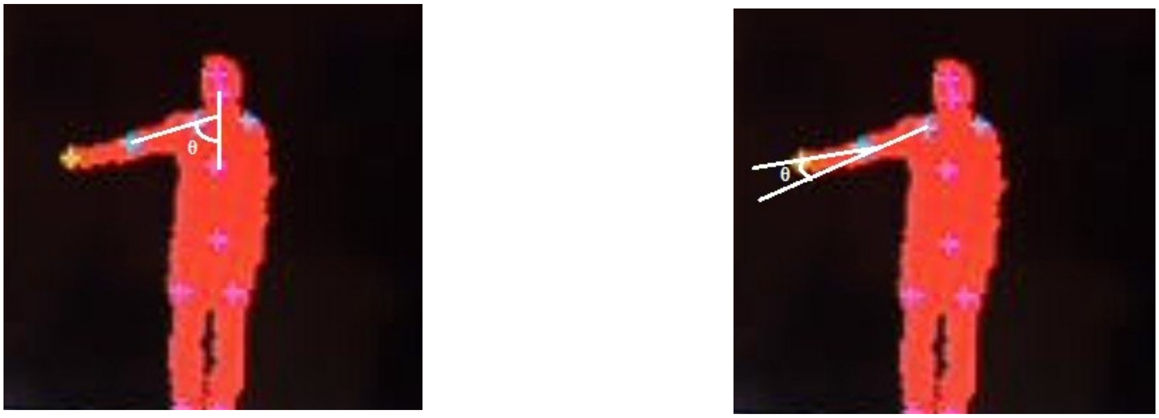
$$u_i = \frac{v_i}{|v_i|} \qquad\qquad 1 \le i \le 2$$

Then the angle θ can be calculated using the following equation

$$\theta = \cos^{-1}(u1_x . u2_x + u1_y . u2_y + u1_z . u2_z)$$

Here $u1$ and $u2$ are the unit vectors calculated from the original vectors.

Using the above angle calculation method, angles which are involved in the particular exercise are generated from the joint positions in each frame. In order to minimize errors in angle calculation, joint confidence is also taken into consideration. The confidence of a joint is a percentage measure of how confident the tracking algorithm is on the result of tracking that specific joint. For example, when a joint is occluded, its confidence drops to a lower value. Similarly, confidence drops to a low value when a joint is outside of the sensor's field of view. It is provided as an API by the Beckon engine which can be used in application programming. The range of the joint confidence is between 0-100. After several experiments it was concluded that any joint whose confidence falls below 60 should be considered erroneous, because the joint is not detected properly or it is out of the field of the view of the camera. In the case of an error in the joint location, a value of -999 is sent back to the main application which tells the application that the angle is not calculated due to error in joint detection. This method basically filters out faulty angular data from data sequence. Sample images in Figure 12 are taken from the application which demonstrates that the angles are calculated.



a) Right shoulder angle calculation

b) Right elbow angle calculation

Figure 12 - a) Right shoulder angle calculation using right elbow, shoulder and torso joint locations. b) Right elbow angle calculation using right finger tip, elbow and shoulder joint locations.

In order to test the accuracy of the particular angle, this thesis uses the angular data of both elbow and shoulder captured from 1250 frames while user maintained the pose. In this experiment the user stretched his right arm straight in a horizontal manner for continuously 1250 frames and obtained the angle of elbow and shoulder from each frame. The angles were stored in a file for further processing. Figure 13 and 14 show the elbow and shoulder angle data obtained from 1250 frames.

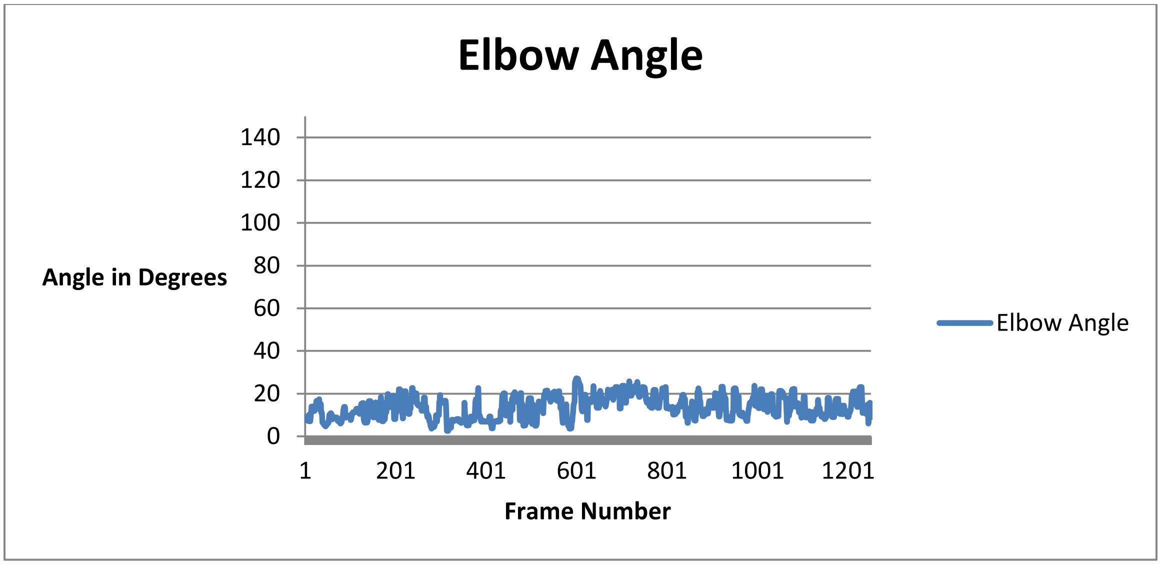


Figure 13 - Elbow angle data obtained from 1250 frames while the user maintains the same pose

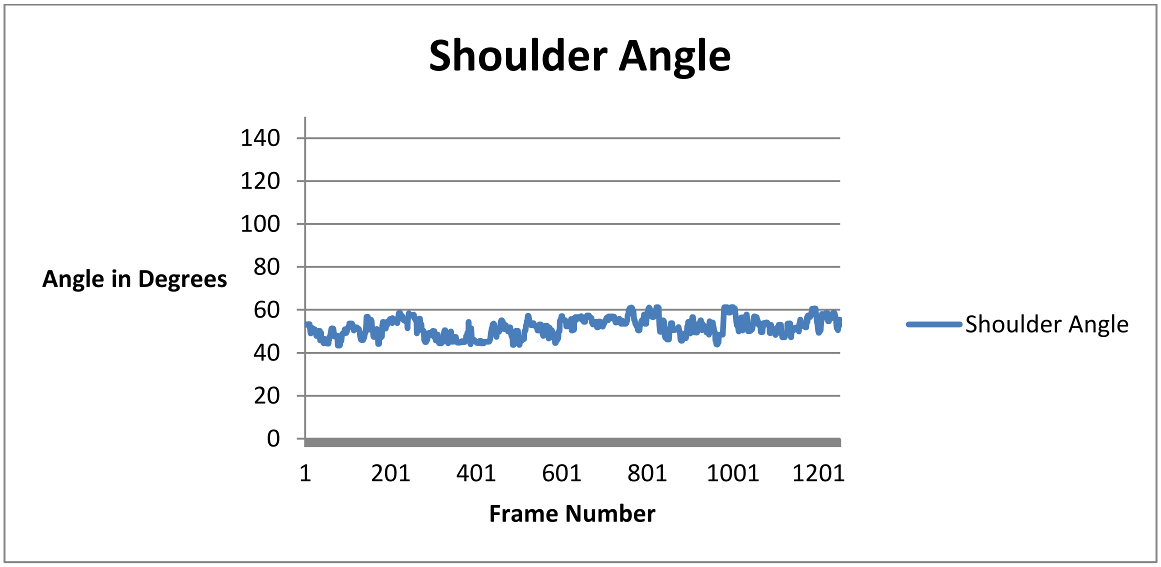


Figure 14 - Shoulder angle data obtained from 1250 frames while the user maintains the same pose

	Frames	**Standard Deviation**
Elbow Angle	1250	5.25
Shoulder Angle	1250	4.23

Table 1 - Elbow and shoulder angle standard deviation obtained from 1250 frames

Table 1 shows the standard deviation of the elbow and shoulder angle data obtained from 1250 frames. The values of standard deviation of elbow and shoulder angle show that the deviation from the average mean angle is not high and this small amount of deviation can be ignored because it does not affect the overall performance of the application.

4.3 Initial and ending gesture

The first challenge here was to define the length of the angle sequences recorded from both the instructor and the patient. In order to overcome this problem an initial and ending gesture for both the patient and the instructor is defined. Standing in front of the sensor, the subject stretches the right arm horizontally. Figure 15 depicts the initial and ending gesture for both the patient and the instructor.

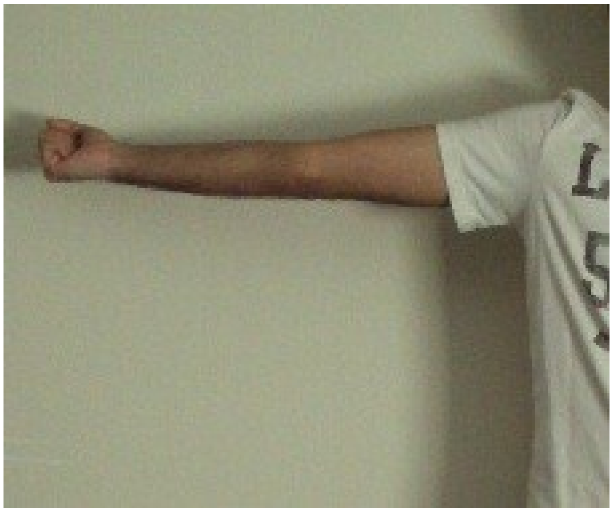


Figure 15 - Right hand stretched in horizontal direction

This gesture lets the application know when to start and stop recording the arm movements of the instructor and patient. A minimum recording threshold of about 50 frames is defined to give both the instructor and the patient enough time to record their sequences. This threshold is defined based on the time the user should be given when he/she stands in front of the sensor with this gesture. The 50 frames threshold equates to a time of approximately 3.25 seconds.

Figure 16 shows the starting and ending of a cycle in an exercise including the initial and ending gesture.

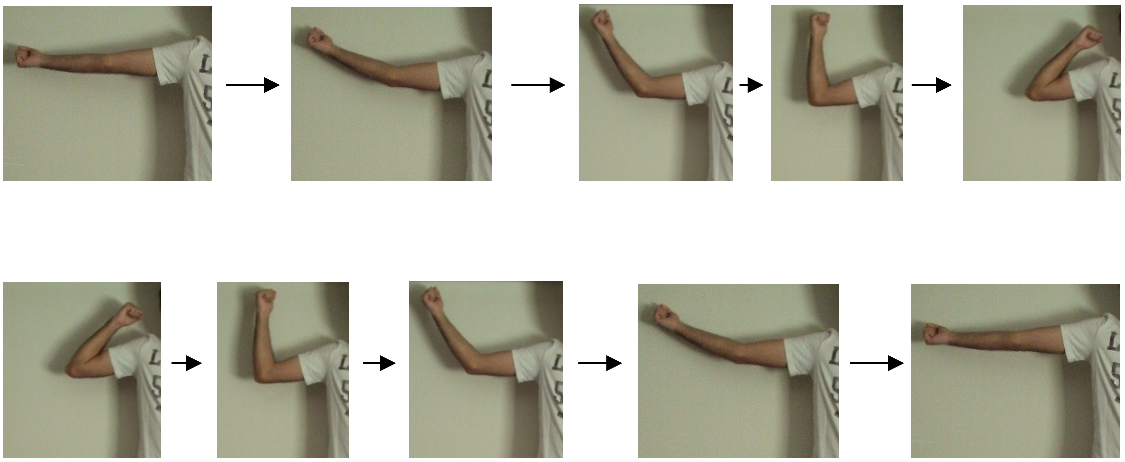


Figure 16 - Right arm cycle including start and ending gesture

Figure 17 and 18 show the sample data of the elbow and shoulder angles of the user captured while performing the above exercise. Curves are not smooth because of the noise in the data. Noise depends on the detection of a joint. Moving out of the sight of sensor or incorrect detection of joint location by the sensor causes the noise. Using joint confidence as discussed earlier is one way to filter noise. But still the joint confidence sometimes provides high confidence for the joints which are detected but the location is not correct. In this case, a median filter algorithm is used to filter this kind of noise. The resultant curves as shown in Figure 17 and 18 are still not completely smooth because some of the data points are discarded by the median filter algorithm. As these filtering techniques apply to both patient's and instructor's data so it does not affect the performance of the application.

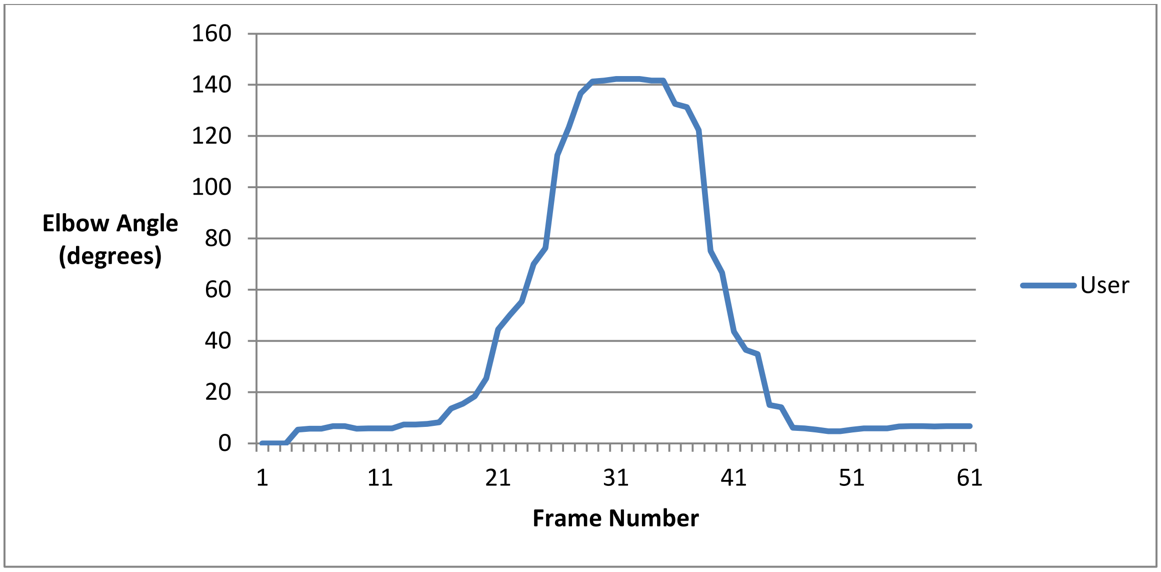


Figure 17 - Elbow angle sequence of a user

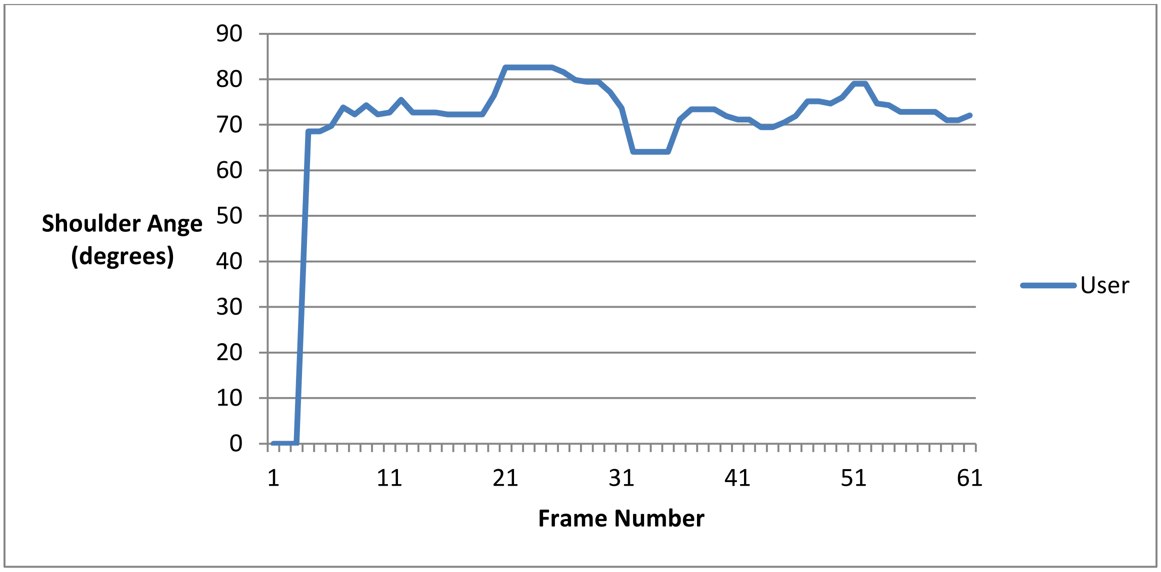


Figure 18 - Shoulder angle sequence of a user

4.4 Median filter

In order to improve the smoothness of the data, data filtering functionality is also implemented. In this part the whole sequence of data is segmented and any noise or jerks caused by incorrect detection of joint is removed. To accomplish this, a median filter algorithm [16] is implemented. The detailed explanation of the algorithm is described below:

The median filter runs through the data sequence entry by entry, replacing each entry with the median of neighboring entries. This thesis uses the window size of 7 while calculating the median at each index. At each index *i* it sorts the elements in the window where *i* is in the center of the window. Once the elements are sorted, the value at index *i* takes on the median of the seven values.

…	*i-3*	*i-2*	*i-1*	*I*	*i+1*	*i+2*	*i+3*	…

4.5 Curve extraction

To handle multiple curves in a single sequence, a curve extraction method is also implemented. The main purpose of this method is to extract the actual curve data from the angular data sequences. Once the instructor and patient start and finish their recordings it is likely that redundant or useless data exists before or after the actual angular curve in each motion data sequence. Another possibility is that either the patient or the instructor might perform multiple exercises in a single sequence. Curve extraction obtains the largest curve from the sequence and provides it to the DTW algorithm to compute the total score.

Figure 19 shows an actual curve enclosed by dotted black lines in an angle data stream.

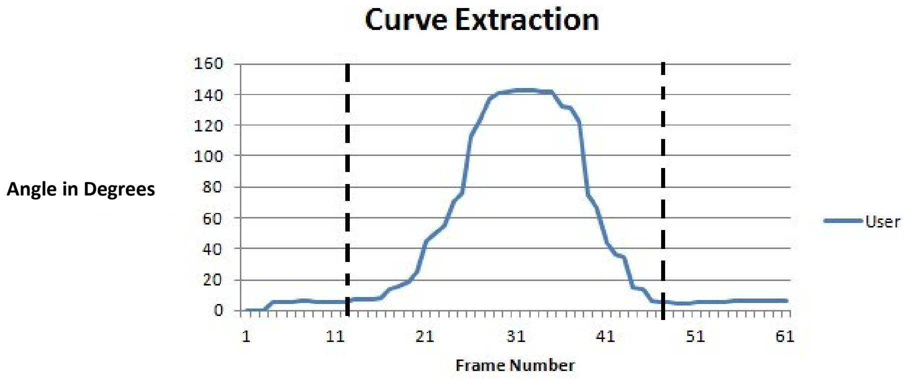


Figure 19 - Actual curve enclosed by dotted black lines

When the user makes the initial or ending gesture in front of the sensor, the value of the elbow angle is calculated between 0 and 8 degrees, and the value of shoulder angle is calculated between 70 and 80 degrees by the application. So in curve extraction method, application performs a linear search of angle data sequence and whenever it finds a value greater than 8 degrees in the elbow angle data sequence it stores the index as a starting point of curve. After that when it finds a value lower than 8 degrees, it means the curve has finished at this point, and then the application calculates the length of the curve by subtracting the index value stored previously from the latest index value. If the application finds another curve within the same data sequence, it compares the length of this curve with the previous one, if the value is smaller than the previous one it keeps the previous one otherwise it replaces the previous curve length and start index with the current curve length and start index. The same method is applied to the shoulder angle data sequence by searching for values which lies between 70 and 80 degrees.

4.6 Dynamic time warping algorithm

In order to compare and compute the distance between two sequences DTW (Dynamic Time Warping) algorithm is used. The complexity of this algorithm is around *O(n*m)* where n is the total number of entries in first data sequence and m is the total number of entries in the second data sequence.

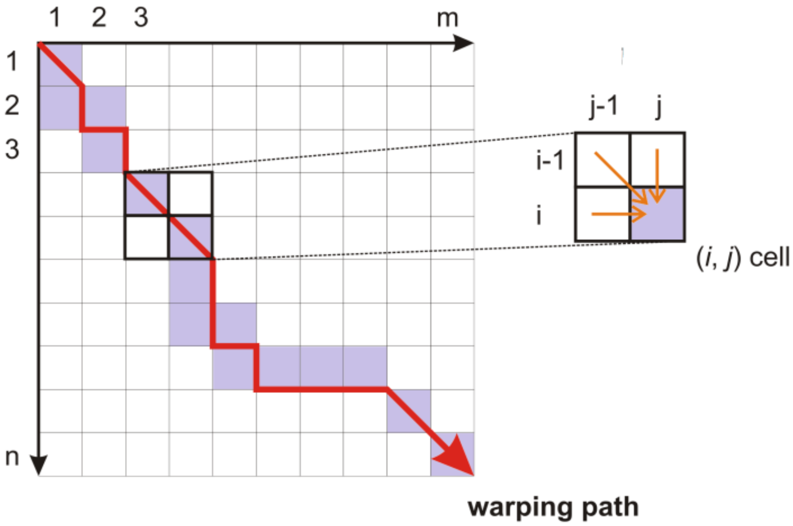


Figure 20 - Dynamic time warping sequence matching [23]

Initially a matrix of size *O(n*m)* is constructed. Location $d(0,0)$ is initialized to 0 and the cells in the first row and the first column of the matrix are initialized to ∞.

$$d(i,0) = \infty \qquad 1 \le i \le n$$

and

$$d(0,j) = \infty \qquad 1 \le j \le n$$

To compute the values of the rest of the cells, the following formula is used

$$d(i,j) = diff(i,j) + \min(d(i-1,j), d(i,j-1), d(i-1,j-1)) \tag{1}$$

Where $diff(i,j)$ is the Euclidean distance between the points at position i and j in both of the sequences.

$$diff(i,j) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (2)$$

After computing all of the cell values, the score in cell $d(n,m)$ is the final score. This final score defines the difference between the two sequences. If the score is low, the sequences are nearly the same, while high scores indicate large differences.

Once the patient is done with his/her exercise, a score will be displayed and the patient will be asked to perform the same exercise again. Both the instructor and the patient are expected to perform a single cycle of an exercise. If either the instructor or the patient performs multiple cycles in one run then the largest curve from both sequences will be extracted to compute the score. This curve by curve comparison is implemented to filter out redundant data from both of patient's and instructor's data sequence, because if that redundant data is passed as it is to the DTW algorithm, it will affect the overall score of the exercise which will make the score less reliable.

4.7 Total score calculation

After extracting the curves from each angular sequence of both the instructor and the patient, the DTW algorithm will compute the score. Once all the scores are obtained they will be added to compute the total score.

$$Total\ Score = \sum_{i=1}^{n} DTW(X_i, Y_i) \qquad \text{where n is number of angles} \quad (3)$$

Here X_i is the ith angle sequence of the instructor and Y_i is the ith angle sequence of patient.

In order to make the score more meaningful, this total is then mapped to a range between 0 and 1. For this purpose, a threshold of 5000 is defined. This value was determined empirically through experiments. Exercise which is performed very poorly results in a score of 5000 or more. If the difference exceeds this limit, it is mapped to a 0, representing very poor performance. The formula in Equation 4 defines calculation mechanism.

$$Score = \begin{cases} 0 & if\ TotalScore > 5000 \\ 1 - \dfrac{TotalScore}{5000} & if\ TotalScore < 5000 \end{cases} \tag{4}$$

CHAPTER V

EXPERIMENTAL RESULTS

In this part of the thesis, experiments were conducted to test the stability, efficiency and correctness of the system. Several patients participated in these experiments. Furthermore, the system's performance was also compared with the high end devices like desktop computer and VICON motion capturing system.

5.1 Multiple patients experiment

In this experiment, two patients participated along with the instructor. The instructor performed the actual exercise first and then both of the patients performed the exercise one by one. Table 2 shows the profiles of the patients who have participated in this experiment.

	Age	**Weight (In Pounds)**	**Gender**	**Height (cm)**
Instructor	25	133	Male	178
Patient 1	25	135	Male	178
Patient 2	34	130	Male	140

Table 2 - Patient profiles

In this experiment, initially an instructor or physical therapist recorded an exercise sequence. Afterwards Patient 1 performed the exercise and recorded his sequence and then Patient 2 performed the exercise and recorded his sequence.

Figure 21 and 22 show the elbow and shoulder angular data of the instructor and both of the patients.

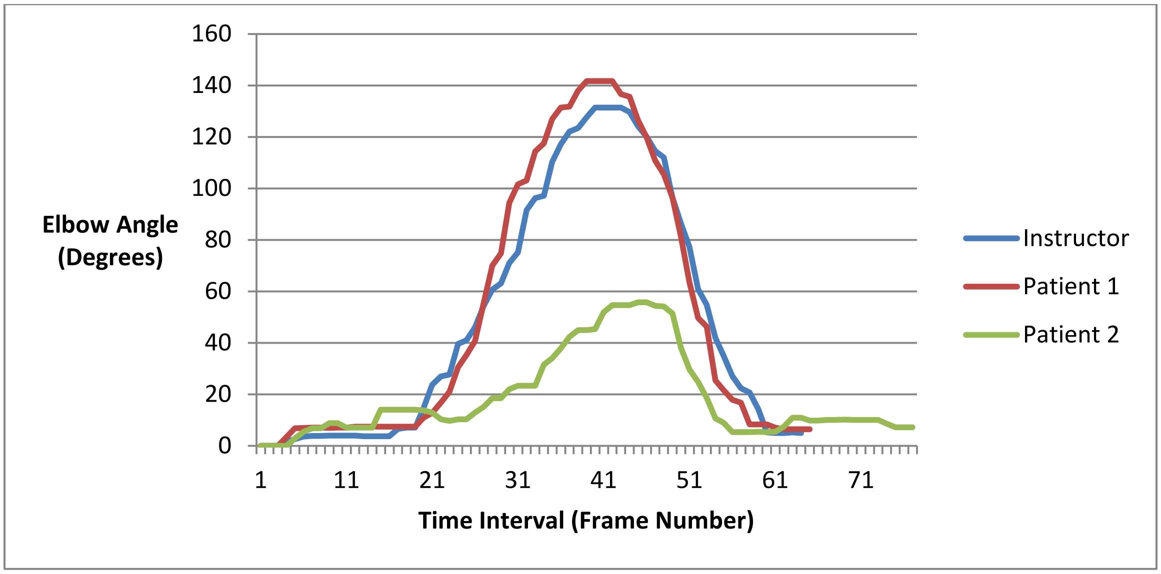


Figure 21 - Elbow angle of Instructor, Patient 1 and Patient 2

The jerk in the curves shows the noise in the data. This noise mostly depends on the incorrect detection of a joint, as well as the distance between the subject and the depth sensor, and it also depends on the background of the subject. Furthermore, the median filter algorithm and the noise filtering technique using joint confidence are already applied to the data shown in Figure 21 and 22. So the jerks in the curves are all because of the missing data which is filtered out by the algorithms mentioned above. These jerks do not affect the comparison score between the curves because the algorithms used to filter out noisy data are applied to each data angle sequence.

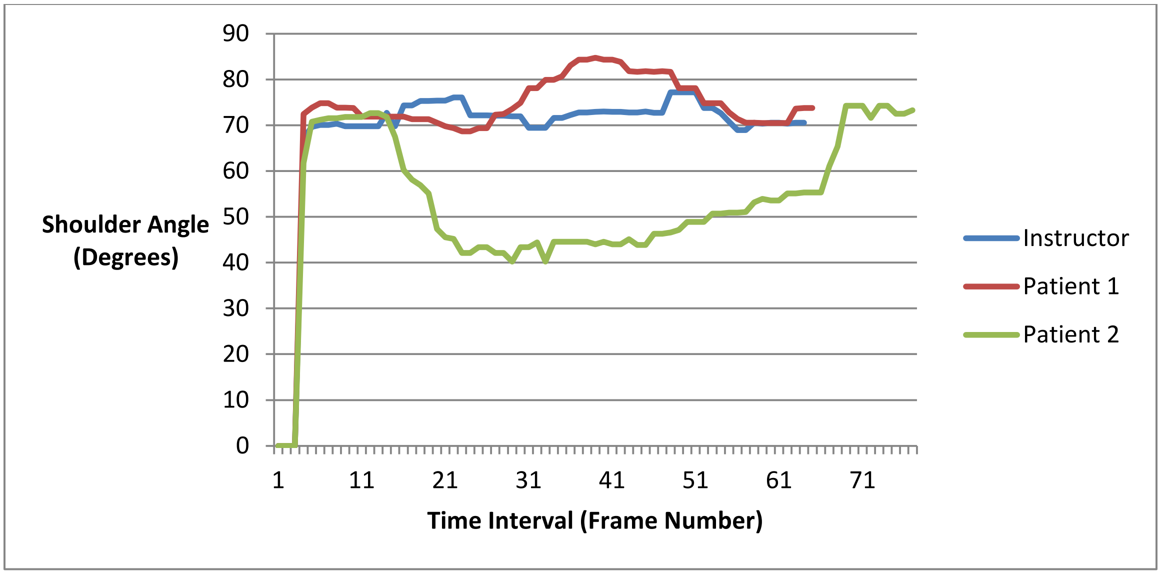


Figure 22 - Shoulder angle of Instructor, Patient 1 and Patient 2

The final score was then computed after comparing the elbow and shoulder angle sequences of the patients with the instructor sequence using DTW and then deriving the score as explained in section 4.8. Table 3 shows the combined score of Patient 1 and Patient 2 calculated using the equation in Equation 4.

	Score
Patient 1	0.87
Patient 2	0.31

Table 3 - Patient 1 and Patient 2 final scores

Here the higher score of Patient 1 depicts that this patient has performed the exercise more accurately (according to the standards of the instructor) as compared to the exercise performed by Patient 2.

In another experiment five patients participated and performed the exercise. Table 4 shows the profile of five patients.

	Age	Weight (In Pounds)	Gender	Height (cm)
Instructor	25	133	Male	178
Patient 1	35	121	Male	155
Patient 2	25	135	Male	178
Patient 3	29	169	Male	176
Patient 4	28	170	Male	177
Patient 5	26	134	Male	175

Table 4 - Five patients profile

Figure 23 shows the elbow angle chart of the instructor and the five patients and Figure 24 shows the shoulder angle chart of the instructor and the five patients.

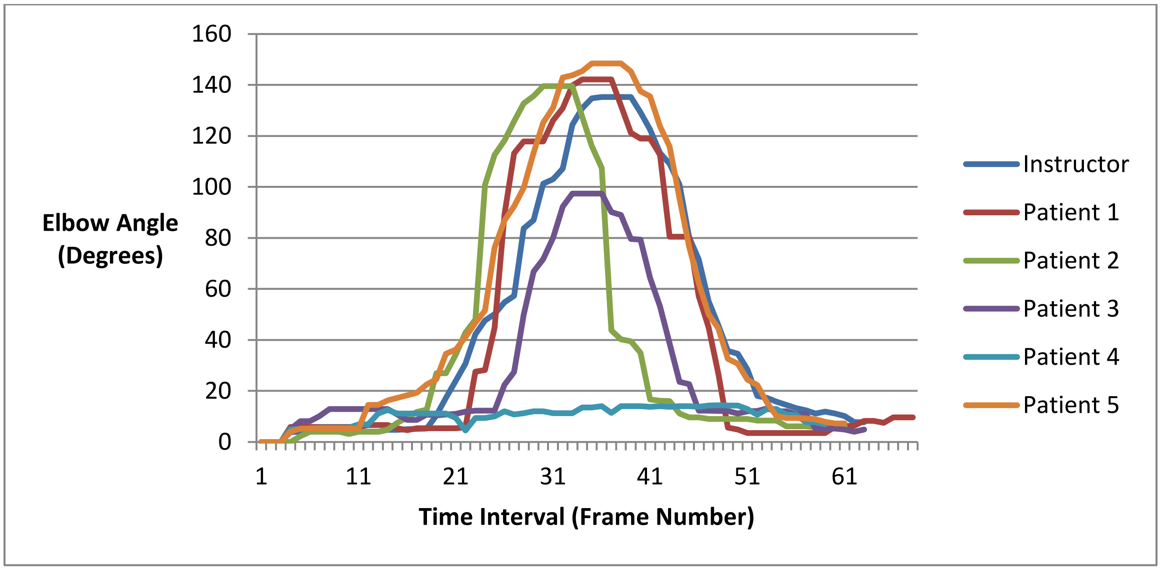


Figure 23 - Elbow angle of Instructor and five patients

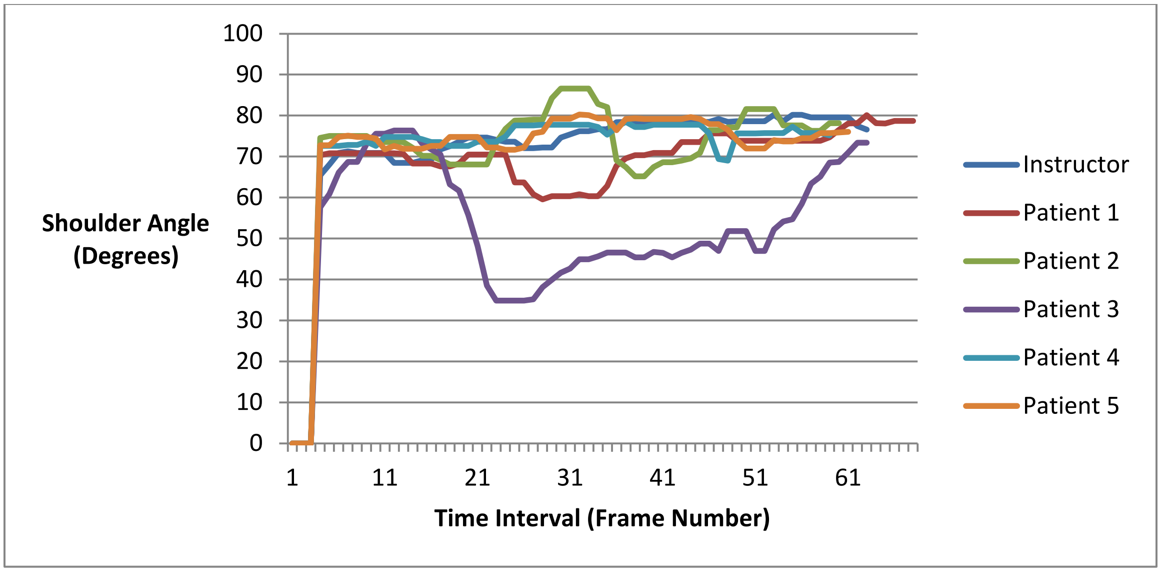


Figure 24 - Shoulder angle of Instructor and five patients

Table 5 shows the score of all five patients. The scores are calculated using the formula in Equation 4.

	Score
Patient 1	0.85
Patient 2	0.86
Patient 3	0.54
Patient 4	0.49
Patient 5	0.90

Table 5 - Final score of five patients

Based on the scores mentioned in Table 5, it is visible that Patient 5 has performed the best of all patients and Patient 4 has performed the worst of all the patients.

5.2 Desktop and Beagleboard FPS comparison

Furthermore, in order to test the efficiency, correctness and stability of the system, a very similar application was created on a desktop machine to calculate the angles. The Beckon SDK for Windows was used for this purpose. The chart below shows the detailed hardware comparison of the desktop machine and the Beagleboard XM on which the actual physical rehabilitation system is running.

	Processor	**Memory**	**Graphics**	**OS**
Desktop	Intel Core i7 2.93 Ghz	4 GB	1.45 billion polygons per second	Win 7 64-bit
Beagleboard	ARM Cortex A8 1 Ghz	512 MB	80 million polygons per second	Angstrom Linux

Table 6 - Hardware and software comparison between Beagleboard and Desktop machine

Table 6 shows that the actual desktop machine has much higher hardware and software capabilities as compared to the Beagleboard.

Figure 25 shows the frames per second (FPS) comparison of both machines.

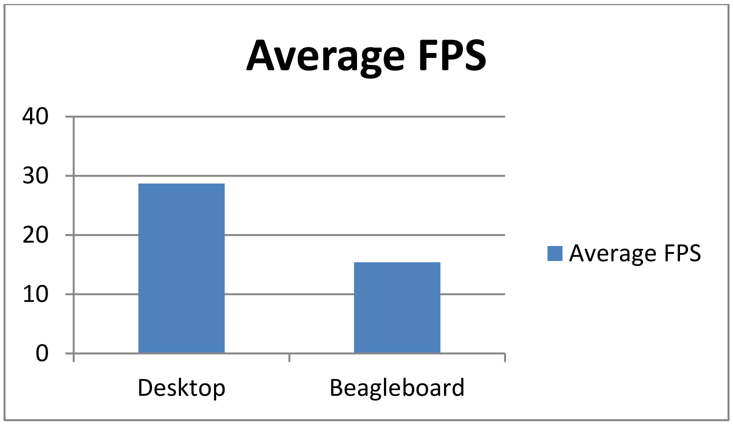


Figure 25 - FPS comparison of Desktop and Beagleboard

This FPS data was collected after executing 700 frames on both machines. The average FPS on desktop was approx 28.6 frames per second whereas the average FPS on Beagleboard was around 15.4.

5.3 Error percentage between Desktop and Beagleboard

To test the stability of the Beagleboard as compared to the desktop machine, elbow and shoulder angular data from 700 frames were recorded from both devices. Figure 26 shows the error percentage, the percentage of frames rejected due low joint confidence or noise.

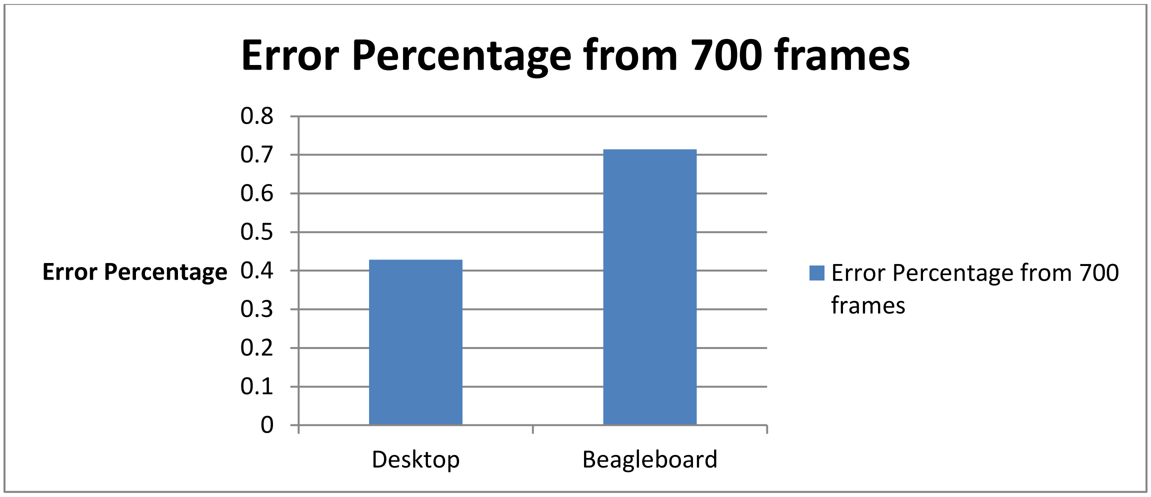


Figure 26 - Error Percentage on both of machines from 700 frames

As Figure 26 shows, the desktop machine's error rate is 0.3% better than that of the Beagleboard. This error percentage in Beagleboard is negligible because the noise filter algorithms have already taken care of these errors. Furthermore, the data sequence is quite large in size so 0.3% error does not affect the overall results when the data sequences are compared.

5.4 Performance and data comparison between Desktop and Beagleboard

Another experiment was performed where both hardware setups were executed in parallel. Both machine's Xtion depth sensors were placed in the same position, one a top the other, so as to have the same view as shown in Figure 27.

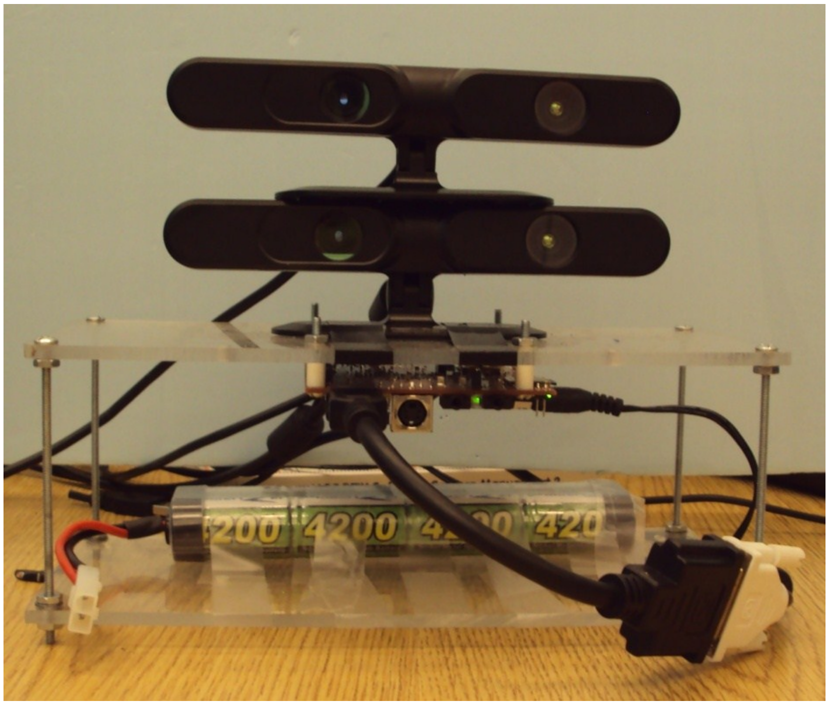


Figure 27 - Desktops's and Beagleboard's Xtion depth sensors are placed in the same position, one a top the other, so as to have the same view.

The subject performed a single exercise cycle and elbow angle data was retrieved from both machines using the common time stamps. Figure 28 shows the elbow angle cycle captured on both machines.

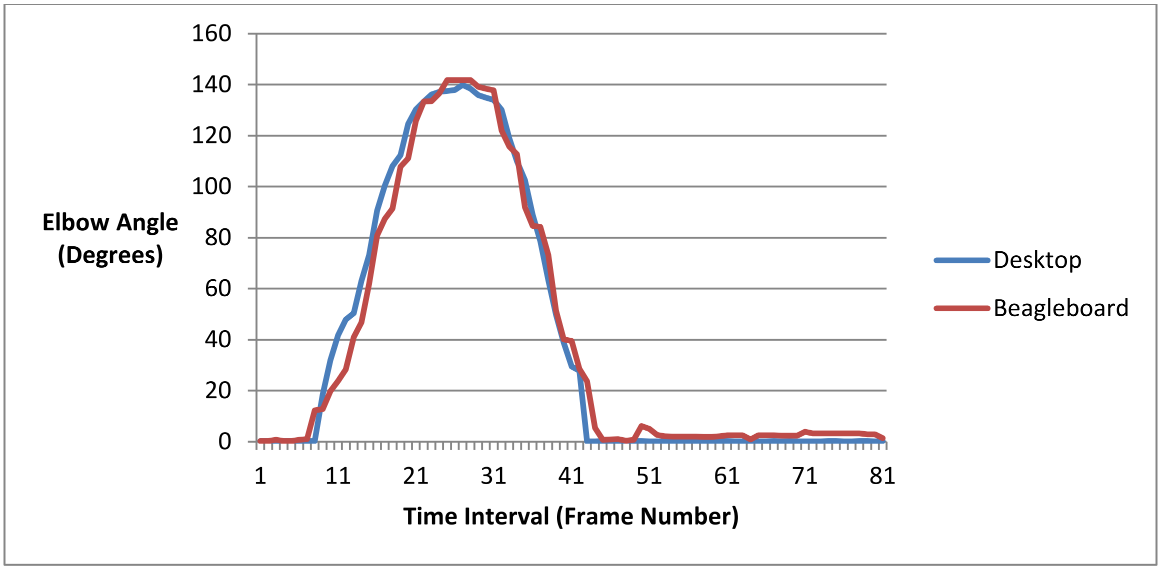


Figure 28 - Elbow angle curves captured from Desktop and Beagleboard (first experiment)

The actual curve captured from the desktop machine was greater in length as compared to the length of the curve captured from the Beagleboard because of the difference of frames per second (FPS) on both machines. So to make the comparison meaningful between desktop and Beagleboard curves, time stamps were used. Using the time stamp, the start and end point of each exercise recording was then determined for both machines. As desktop machine's FPS is almost twice of the FPS on Beagleboard, each alternative entry in the desktop curve was then removed. By doing this the length of the curves becomes almost the same while keeping the original data intact. After that either of the data sequence curve was then padded manually so to adjust it to the starting point of the other curve. The main reason of this padding was because of the difference of milliseconds in both of the machine's timestamps.

Once the data was captured from both machines, the curves were compared using the same DTW (Dynamic Time Warping) algorithm used to evaluate patient's performance. The algorithm returned a difference of 233.364 points. Mapping it between the range of 0 and 1 using the formula in Equation 4 yields 0.95, which is a very good score.

In another experiment, again elbow angle curve data was captured from both the desktop and the Beagleboard. Figure 29 shows the results after down sampling the results of desktop machine.

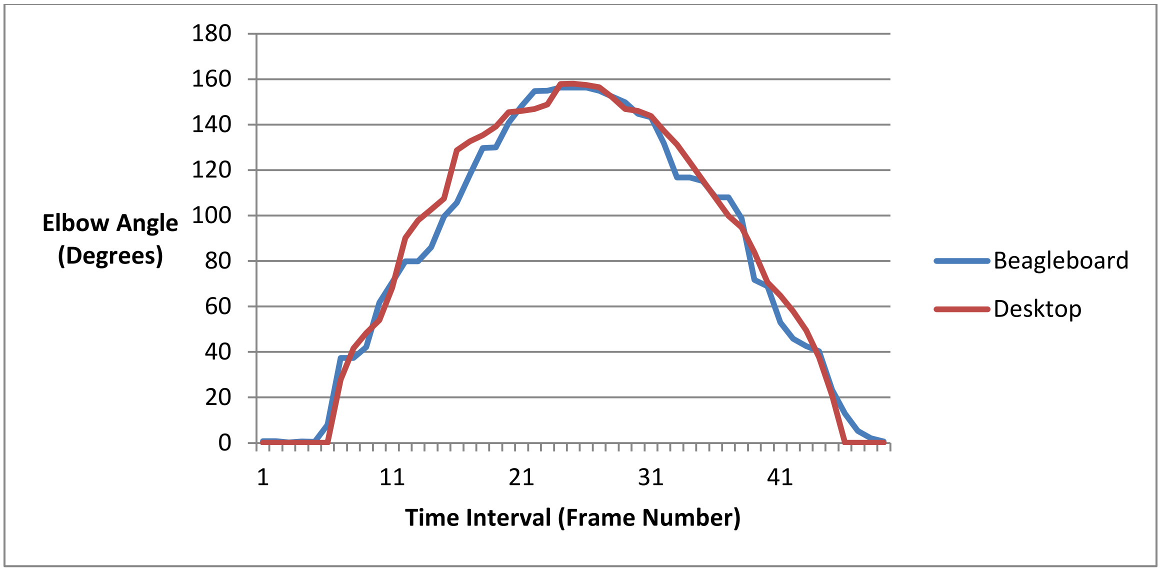


Figure 29 - Elbow angle curves captured from Desktop and Beagleboard (second experiment)

Table 7 shows the results of both of the experiments:

	Score
Experiment 1	0.95
Experiment 2	0.96

Table 7 - Experiments Result

These experiments show that even though the Beagleboard has very limited hardware capabilities as compared to the desktop machine, the performance is quite similar.

5.5 Performance comparison between VICON and the Physical Rehabilitation System

In this test 12 markers were attached to the user's body at the position of right hand finger tip, right hand elbow and right shoulder, as shown in the Figure 30

Figure 30 - Twelve markers attached to right hand finger tip, elbow and shoulder for detection by the VICON motion capture system

Both systems, the VICON motion capture system and the Beagleboard, ran in parallel. The user performed a single right hand cycle. Both systems captured the joint positions of the right hand finger tip, elbow and shoulder and calculated the elbow angle throughout the motion. The capture rate of the VICON was around 100 frames per second whereas the capture rate of Beagleboard was around 15 frames per second. So in order to compress the elbow angle curve data captured from VICON to match with that of the Beagleboard, the same methodology discussed in section 5.5. First the data was collected from both the machines using the common timestamps. As the VICON's FPS is almost 6.5 times higher than the FPS of Beagleboard, every sixth frame was taken into consideration. After that either of the curves was then padded manually to adjust the starting point of other curve because of the difference of milliseconds in timestamps. Figure 31 displays the elbow angle curves captured by both systems using the common time stamps. Each point in the VICON elbow angle curve displayed in the graph below represents those points from the actual data which are separated by the distance of 6 frames, In other words, every sixth frame data is taken into consideration.

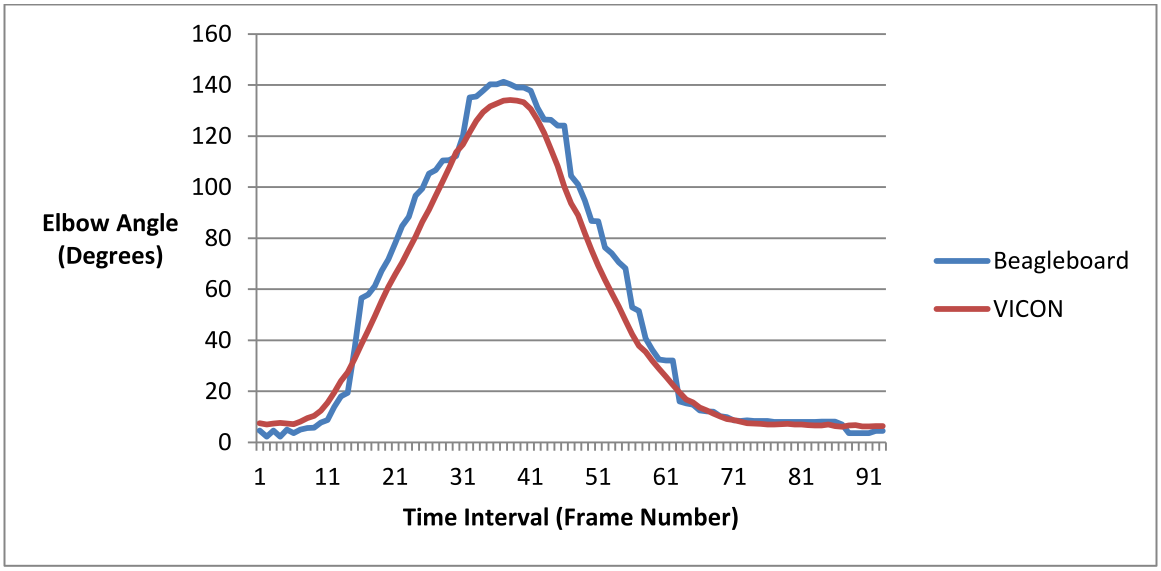


Figure 31 - Elbow angle data captured from VICON and Beagleboard

The distance between these two curves was again calculated using the DTW algorithm. Using the formula in Equation 4, the final score was 0.94 between the VICON and the Beagleboard curves. This experiment shows that the performance of the low cost, portable Beagleboard is similar to the performance of an exercise laboratory solution like the VICON motion capturing system, in terms of the therapy evaluation task.

CHAPTER VI

CONCLUSIONS

In this thesis a low cost low power Xtion depth sensor based system is presented which provides patients the means to evaluate performance of their physical exercises, as instructed by their physical therapists. Research in the domain of physical rehabilitation has used high end machines and laboratory settings to execute their applications, whereas this application runs on low-cost, portable hardware. The system uses the Asus Xtion sensor to capture human activity and the Beagleboard to process the data. Other hardware components include a battery and a LCD touch screen device. All of the hardware components involved are cheaply available in the market which makes the price of the overall system substantially less than other approaches. On the software side the Dynamic Time Warping and Median Filter algorithms are used to compare an instructor's and patient's angular sequences calculated from joint positions. These algorithms are implemented efficiently to minimize the processing and memory requirements.

Furthermore, several experiments were conducted on patients of different profiles after an instructor recorded the original sequence. The patient who performed badly received a low score and the patient who performed well received a good score. These scores suggest the correctness of the system. Because a few numbers of experiments were conducted in this regard, this also increases confidence in the reliability of the system. To evaluate the performance of the system when compared to high end machines such as a desktop machine or VICON motion capturing system, several experiments were conducted where the same angle calculation algorithm ran in parallel on both machines. These experiments showed that the low-cost system has performance almost identical to laboratory hardware solutions.

Using the results of above experiments, it can be concluded that despite slow and limited memory hardware, the system performs the physical therapy evaluation task at a level similar to that of high end machines.

CHAPTER VII

FUTURE WORK

This thesis described a system which can capture and evaluate a patient's exercises based on a physical therapist's demonstration, using low cost low power, portable hardware. The current research focuses on using two angles, the right elbow and shoulder angles. The system can be easily expanded by including more angular sequences. This will enable the system to evaluate more complex exercises like dancing, karate, golf playing etc.

Including more angles will also enable the system to be used in several domains like drowsiness detection for drivers in automobiles, using angles from the joints of the upper human body. It can also be used in robot learning to record robot moves and provide performance feedback to enhance robot learning. Other applications include sports, where it can be used to evaluate the performance of an athlete.

REFERENCES

1) Boato, G.; Conci, N.; Daldoss, M.; De Natale, F. G B; Piotto, N., "Hand tracking and trajectory analysis for physical rehabilitation," *Multimedia Signal Processing, 2009. MMSP '09 IEEE International Workshop on* , vol., no., pp.1,6, 5-7 Oct. 2009

2) Ruzena Bajcsy and others "Upper Extremity Reachable Workspace Evaluation with Kinect", *University of California at Berkeley, 2013*, URL: http://www.researchgate.net/publication/235522326_Upper_extremity_reachable_workspace_evaluation_with_kinect

3) Jun-Da Huang "KinereHab: A Kinect Based System for Physical Rehabilitation", ASSETS '11 *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, Pages 319-320, New York, USA 2011

4) Leandro Miranda, Thales Vieira, Dimas Martinez, Thomas Lewiner, Antonio W. Vieira, Mario F. M. Campos "Real time gesture recognition using key poses and depth data", *UFMG, 2012*, URL: http://sibgrapi.sid.inpe.br/rep/sid.inpe.br/sibgrapi/2012/07.06.21.52

5) Luc Geurts, Vero Vanden Abeele, Jelle Husson, Frederik Windey, Maarten Van Overveldt, Jan-Henk Annema, Stef Desmet "Digital Games for Physical Therapy", *TEI '11 Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, Pages 117-124, ACM New York, NY, USA 2011

6) Samyukta Ganesan and Lisa Anthony "Using the Kinect to Encourage the Older Adults to Exercise", *University of Marlyland Baltimore County, 2012*, URL: http://kinectforexercise.files.wordpress.com/2012/02/ganesan-anthony-chi12-wip.pdf

7) Chien-Yen Chang, Belinda Lange, Mi Zhang, Sebastian Koenig, Phil Requejo, Noom Somboon, Alexander A. Sawchuk, and Albert A. Rizzo "Towards Pervasive Physical Rehabilitation Using Microsoft Kinect", *University of Southern California, May 2012*, URL: http://ict.usc.edu/pubs/Towards%20Pervasive%20Physical%20Rehabilitation%20Using%20Microsoft%20Kinect.pdf

8) A. Da Gama, T. Chaves, L. Figueiredo, V. Teichrieb "Poster: Improving Motor Rehabilitation Process through a Natural Interaction Based System Using Kinect Sensor", *3D User Interfaces (3DUI), 2012 IEEE Symposium*, March 2012

9) Andrew D. Wilson and Aaron F. Bobick "Learning Visual Behavior for Gesture Analysis", *Computer Vision, 1995. Proceedings., International Symposium*, Nov 1995

10) David Kortenkamp, Eric Huber and R. Peter Bonasso "Recognizing and interpreting gestures on a mobile robot", *In Proceedings of AAAI-96*, 1996, URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.8823

11) "Omek Beckon SDK DM3730 Edition – Developer's Guide". *Omek Interactive, 2012,* URL: http://support.omekinteractive.com/index.php?/managedownloads/Download/View/34/11/documentation

12) Isaac Pastor Acosta "Upper Limb Rehabilitation of Stroke Patients Using Kinect and Computer Games", *Univeristy of Utah, August 2012*, URL: http://content.lib.utah.edu/utils/getfile/collection/etd3/id/2036/filename/2043.pdf

13) Belinda Lange and others "Interactive Game-Based Rehabilitation using the Microsoft Kinect", *IEEE Virtual Reality 2012*, March 2012

14) Chuan-Jun Su and others "Ensuring Home-based Rehabiliation Exercise by using Kinect and Fuzzified Dynamic Time Warping Algorithm", *Asia Pacific Industrial Engineering & Management Systems Conference 2012*

15) Daniel Lamire "Faster Retrieval with a two-pass Dynamic Time Warping lower bound", *Pattern Recognition* 42(9): 2169-2180 (2009)

16) E. Arias-Castro and D.L. Donoho, "Does median filtering truly preserve edges better than linear filtering?", *Annals of Statistics*, vol. 37, no. 3, pp. 1172–2009

17) "BeagleBoard-xM page". September 14, 2010. Retrieved Sep 12, 2012. URL: http://beagleboard.org/hardware-xM

18) “VICON page”, Retrieved May 10, 2013, URL: http://www.vicon.com/

19) “Optitrack motion capture page”, Retrieved June 23, 2013. URL: http://www.naturalpoint.com/optitrack/

20) Microsoft (2012). "Kinect for Windows SDK 1.6 Programming Guide". *Microsoft.* Retrieved February 16, 2013.

21) “Asus Xtion Pro page”, Retrieved June 23, 2013. URL: http://www.asus.com/Multimedia/Xtion_PRO/

22) "A Brief History of Qt", Retrieved December 20, 2007. URL: http://my.safaribooksonline.com/0131872494/pref04?portal=oreilly

23) “Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining”, Retrieved June 25, 2013. URL: http://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/similarity-measures-and-dimensionality-reduction-techniques-for-time-series-data-mining

24) “BeagleBoard-xM Development Board“. Retrieved June 25, 2013. URL: http://www.ti.com/tool/beaglexm

25) “Xtion Pro”, Retrieved June 25, 2013. URL: http://www.asus.com/Multimedia/Xtion_PRO

26) “Transcend 8GB SD Card”, Retrieved June 25, 2013. URL: http://www.naaptol.com/memory-cards/transcend-8gb-sd-card/P/746492.html

APPENDICES

APPENDIX A

Adding Runtime Graph plotting in QT

This is a tricky part. The first challenge was to find the correct library for the QT. After a thorough search I was able to find a library named as Qwt using which we can plot graphs in QT interfaces. Now the next challenge was to configure and add this library into the QT running on embedded system like Beagleboard. Qwt is not properly configured to run on embedded systems. So in order to do that first you have to download the Qwt from internet. The packages available could have name like “qwt-6.1-rc3” etc.

Extract the files from package and open the Qwtconfig.pro in QT IDE. Disable options like Designer and OpenGL by commenting them and save it. Build the project using qmake for ARM. It can be done either manually or you have to configure it after installing the QT Creator in Ubuntu machine. There is a tutorial mentioned in Omek Beckon SDK for Beagleboard developer’s guide about installing QT Embedded SDK for Angstrom/Beagleboard on a linux machine. It is available on the following link http://support.omekinteractive.com/index.php?/managedownloads/Download/View/34/11/documentation

Once the Qwt project is successfully built in the QT Creator, a folder will be created on the target build path. Proceed to that folder in the terminal of your linux machine and execute the following commands.

```
$ make
$ make install
```

These commands will install the Qwt libraries into the /user/local/qwt-xxx/include and /user/local/qwt-xxx/lib/ paths of the linux machines.

Now the next part is to add these library references into the actual project. This can be done by adding references into the .pro file of the QT project. These are the lines which should be added into the .pro file of the project.

```
IncludePaths += /usr/local/qwt-xxx/include
LIBS += /user/local/qwt-xxx/lib/libqwt.so.6
```

Now the Qwt graph functionalities are available in the project. After writing the code for the graphs, build the project and copy the project executable into the destination directory of Beagleboard. Also copy the libqwt.so.6 from the path mentioned into the same destination directory of the Beagleboard where the project executable has been placed. This will now enable the actual project file to use the Qwt plotting libraries on the Beagleboard platform.

VITA

Usman Zafar

Candidate for the Degree of

Master of Science

Thesis: EVALUATING PHYSICAL REHABILITATION WITH A KINECT BASED LOW COST SYSTEM

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2013.

Completed the requirements for the Bachelor of Science in Computer Science at National University of Computer & Emerging Sciences, Lahore, Pakistan in 2010.

Experience:

Graduate Research Assistant
Oklahoma State University, Stillwater, OK *Jan 2012 to present*

- Working on thesis titled "Evaluating Physical Rehabilitation with a Kinect based low cost system"
- Also worked on various research projects

Software Engineer
MWM Ventures, Lahore, Pakistan *Oct 2011 to Dec 2011*

- Worked on a newer version of mobile based music application known as UrFilez

Software Engineer
Strategic Systems International, Lahore, Pakistan *Jun 2010 to Oct 2011*

- Worked on My Print Cloud named commercial product which provides cloud based solution to manage printing activities
- Designed and developed the additional modules of the SmartHR named enterprise level application for the internal client

Professional Memberships: IEEE-NUCES, ACM