BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Activity Recognition of a Badminton Game Through Accelerometer and Gyroscope

Authors

Md. Ariful Islam Anik	Id: 124412
Mehedi Hassan	Id: 124426

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
Organisation of Islamic Cooperation (OIC)
Gazipur-1704, Bangladesh

November, 2016

Systems and Software Lab

Islamic University of Technology

Department of Computer Science and Engineering (CSE)

Activity Recognition of a Badminton Game Through Accelerometer and Gyroscope

Authors

Md. Ariful Islam Anik	Id: 124412
Mehedi Hassan	Id: 124426

Supervisor

Dr. Kamrul Hasan
Associate Professor
Department of CSE
IUT

A thesis submitted to the Department of CSE

in partial fulfillment of requirements for the degree of B.Sc. Engineering in CSE

Academic Year: 2015-16

November - 2016

Systems and Software Lab

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Md. Ariful Islam Anik and Mehedi Hassan under the supervision of Dr. Kamrul Hasan, Associate Professor of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Md. Ariful Islam Anik

Student ID - 124412

Mehedi Hassan

Student ID - 124426

Supervisor:

Dr. Kamrul Hasan

Associate Professor

Department of Computer Science and Engineering

Islamic University of Technology (IUT)

Acknowledgement

We must offer our sincerest gratitude towards the individuals without whom this thesis would not be possible by any means. We begin with our honorable supervisor **Dr. Kamrul Hasan** Sir, Associate Professor, Department of CSE, IUT for giving us the cherished opportunity to be a part of the Systems and Software Lab(SSL) where we got an experience of true research exposure. He guided us with every aspect of our research and showed us the way when we were at a loss many a times. He also provided the necessary hardware and software materials according to the need of our research. Not only this but he also made it clear to us that his door is always open for us whenever we needed consultation. Another person that must mention is **Hasan Mahmud** Sir, Assistant Professor, Department of CSE, IUT who always monitored our progress and gave us valuable insights into technical matters that are thoroughly related to our research. It was truly a great overall experience to be able to become a part of this journey. It was our pleasure to get the cooperation and coordination from Professor Dr. M.A. Mottalib, Head of CSE Department, IUT during various phases of the work. We are grateful to him for his constant and energetic guidance and valuable advice.

Our deep appreciation extends to all the respected jury member of our thesis committee for their insightful comments and constructive criticism of our research work. Surely they have helped us to improve this research work greatly. Lastly we would like to thank the faculty members of CSE Department, IUT who have helped to make our working environment a pleasant one, by providing a helpful set of eyes and ears when problems arose.

Abstract

The scope for doing physical exercises in daily life is declining day by day. But, the importance of human physical exercise for a healthy life, remains the same. It is necessary to generate a solution to simulate the outdoor experience of physical exercises and sports inside our home. In this paper, we propose an idea of recognizing the activities of a badminton game which has the potential to be useful in simulating the Badminton Sport. We have used motion sensors (e.g. Accelerometer, Gyroscope) to recognize different activities like, serve, smash, backhand, forehand, return etc. We have collected data from a large set of users and labeled their data over several instances. We have applied the Root-Mean-Square(RMS),K-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) and Dynamic Time Warping(DTW) classifiers and to recognize those activities. Existing approaches (e.g. Microsoft Xbox 360) used vision based techniques to recognize activities and use it in simulated games but we are using sensor based approach. Vision based approaches have some limitations such as the slow rate of data, illumination constraints, occluded backgrounds etc. Our approach gives a low cost solution with a classification technique which is faster. The experimental result shows a decent recognition rate.

Keywords: Accelerometer, Gyroscope, Activity Recognition, Physical Exercise.

Contents

List of Figures

Chapter 1

Introduction

Physical activity is any activity that helps to improve or maintain ones physical fitness. For the assurance of a healthy physique and mind there are hardly any substitutes of regular physical exercise and activities. As we speak of activities, they also extend to several outdoor sports and games. But from our current social view point and working schedule it is not possible for the people from all professions to participate in physical exercises and outdoor sports in a regular basis. Studies show that physically active people are less likely to develop coronary heart disease than those who are inactive [1].

1.1 Motivation

So, we decided to focus our research on how can a usual person who doesnt have regular opportunity of physical exercise and sports can have the same experience in his leisure without even stepping out of this room. The initial idea that we encountered is a Virtual Reality game of the Badminton Sport that can give a real life experience. The reason for selecting Badminton among the others sports is that the range of exercises that one can do for the sport of badminton is really quite big. This wide range of physical movements will enable a person to explore various exercise related activities.

1.2 Problems in Existing Solutions

It is noteworthy that there are in fact some existing solutions for the mentioned cause. The well-known providers are Xbox360, PlayStation, Wiii Neaton etc. But these solutions cost a hefty amount of money. They also have some technical issues. Most of these games use a vision based system where low light, ambiguous background, unaware movement might be mistaken as a move of the game itself. The vision based systems are also known to be on the slower side when it comes to the flow of data. In our Virtual Reality based game we want to integrate a sensor based system. Thus it will be immune from all the limitations mentioned above and also the rate of data will be much higher in our case.

1.3 Our Suggested Solutions

Activity recognition is the mathematical interpretation of a human motion by a computing device. In personal computing, gestures are most often used for input commands. Recognizing gestures as input allows computers to be more accessible for the physically-impaired and makes interaction more natural in a gaming environment [2]. In this activity recognition part, we will identify some well-known and mostly used moves of the Badminton game. The selected moves for our system are Smash- a move that generates the highest magnitude of force from a player, Serve- a move to start the game from one end, Return- A move that is played with both hand to reverse the serve of the opponent, Backhand- A move played with rear part of one hand, Forehand- a move played with usual hand position. Recognizing these moves of the game will enable us to involve them in our virtual reality model to bring in the precision so that players get a feel of the real game.

For this entire system, firstly we need a Badminton Racket. This racket will be integrated with a MPU-6050 sensor. An MPU-6050 is a conjunction of an Accelerometer and a Gyroscope. An Accelerometer is a device that measures proper acceleration (g-force). Proper acceleration is not the same as coordinate acceleration (rate of change of velocity). For example, an accelerometer at rest on the surface of the Earth will measure an acceleration g= 9.81 m/s2 straight upwards. By contrast, accelerometers in free fall (falling toward the center of the Earth at a rate of about 9.81 m/s2) will measure zero. A Gyroscope is a spinning wheel or disc in which the axis of rotation is free to assume any orientation by itself. The Racket along with the MPU-6050 will be connected to an Arduino module. This Arduino module creates a data flow with the computer. The accelerometer will provide us with raw Acceleration data in the X, Y, and Z axes and the gyroscope will provide us the angular value in the X, Y and Z axes. The Racket along with the MPU-6050 will be connected to an Arduino module. This Arduino module creates a data flow with the computer. The accelerometer will provide us with raw Acceleration data in the X, Y, and Z axes. The successful implementation of this idea will lead us to a phase where the collected results from the algorithms implemented can be used for training Artificial Intelligence components. For example if someone is playing a virtual reality badminton game against the CPU, the human user need not be told about the prcised aspects of every move. But the CPU has to be trained with the data for getting life like results as it is an AI component.

As far as the Activity Recognition idea is concerned, accelerometer will provide the acceleration data. This data will be segmented, classified and the features of the data will be extracted in our proposed model. We will implement the idea based on RMS values, K-Nearest Neighbor (k-NN) classifier and Clustering based approaches along with the Support Vector Machine (SVM) and Dynamic Time Warping (DTW) approach.

Furthermore, the in-depth elaboration of the Architecture for the Activity Recognition System and scope of its future usability follows next in this paper with related works that have already been done in this particular sector of Activity Recognition of the Badminton sport.

Chapter 2

Related Works

Our goal is to correctly recognize activities performed in a badminton game using motion sensors. Also incorporate this recognized activity in a virtual reality game. The game will be controlled by a sensor enabled badminton racket. Therefore, the key things here are activity recognition from sensor value and the movement of a sensor enabled badminton racket. Lots of work have been done in the field of activity recognition.

2.1 The Basics of Activity Recognition

Activity recognition is an important technology in pervasive computing because it can be applied to many real-life, human-centric problems such as elder care and healthcare.Successful research has so far focused on recognizing simple human activities.The goal of activity recognition is to recognize common human activities in real life settings. Accurate activity recognition is challenging because human activity is complex and highly diverse.[3] It aims to recognize the actions and goals of one or more users from a series of observations on the users' actions and their environmental conditions.Current research in the area of activity recognition focuses among other things on personalization, on increasing the number of activities to be recognized. Only few publicly available datasets exist in the research field of human activity monitoring.[4]

2.2 Different Approcaches to Activity Recognition

Accelerometer is a widely used motion sensor for recognizing activities.A triaxial accelerometer is a sensor that returns a real valued estimate of acceleration along the x, y and z axes from which velocity and displacement can also be estimated. The recognition should be done based on some defined activities.The activity recognition algorithm should be able to recognize the accelerometer signal pattern corresponding to every activity.Alternate approaches to activity recognition include use of Hidden Markov Models(HMMs) or regression. HMMs would be useful in recognizing a sequence of activities to model human behavior. Nishkam Ravi, Nikhil Dandekar, Preetham Mysore and Michael L. Littman concentrate on recognizing a single activity. Regression is normally used when a real-valued output is desired, otherwise classification is a natural choice.[5]

The new technologies of health monitoring devices range from on-body wearable sensors to in vivo sensors. For instance, bio-sensors are generally used to monitor vital signs such as electrocardiography (ECG), electromyography (EMG), blood pressure, heart rate and temperature. Illnesses such as seizures, hypertension, dysthymias, and asthma can be diagnosed and treated by physiological monitoring. Inclinometers and goniometers are other types of sensors that are used to measure upper/lower limbs kinematics.[6]

Inertial sensors have been used mainly for navigation of aircraft, ships, land vehicles and robots, and also for shock and vibration analysis in several industries. Rapid development of microelectromechanical systems technology has contributed to the development of small-size,lightweight and low-cost inertial sensors. Currently, many manufacturers propose inertial sensors that are easy to attach and wear. These sensors allow one to collect data on daily living activities under free-living conditions and over extended periods of time. The number and the placement of inertial sensors on the human body have a direct impact on activity recognition, in terms of the variety of activities to monitor and the precision of their classification.[7]

Monitoring and classification of human activity using simple body-worn sensors is emerging as an important research area in machine learning. Activity monitoring itself is motivated by a variety of mobile and ubiquitous computing applications, such as personalisation of the user interface, behavioural monitoring in medicine, medication assessment, assistive systems for the elderly and cognitively disabled or intelligent information delivery and recording systems for industrial assembly and maintenance.In contrast to isolated motion recognition that has been shown in various areas, the spotting task is much more challenging. The difficulty of spotting specific human motion events stems from a number of sources.Although many motion recognition approaches exist, few are dealing with the spotting task itself.Some of them proposed a method for spotting gestures in continuous data.Those approaches make use of an HMM-based accumulation score that supports endpoint detection of a particular gesture in a continuous data stream.[2]

2.3 Recognition of Activities in Other Games

There have been similar works in the field of analyzing strokes and smash in various racket and bat based game. M. Lapinski from MIT Media Lab had created a sensor-based system to analyze baseball movements called SportSemble [8] which is a 6 DOF (degree-of-freedom) sensor node that able to indicate the force and torque signals of the players. It consists both low and high range sensors so it is able to detect any motion from low speed to high speed. By using the IMU, it is possible to present the swing profile of the bat.

There are some research on the tennis movement analysis and commercial product to track the tennis performance. ZEPP Tennis [9] is one of the popular tennis swing tracking sensor which able to track the information of the ball serving, impact, sweet spot, and provide a 3d feedback to the player. Yet the tennis sensor is not suitable to apply on badminton game since the badminton and the tennis game are actually different type of sport even there are some similarities on them. Tennis is a sport that involved the movement mainly on the left and right, the ball is much heavier and much power needed to whack them to the opposite side and the overall speed of the game is slower. While badminton is a sport that involved a lot of footwork on the game field, lighter shuttle weight, consumed greater stamina, highest game speed among the racket sport and it is required a lot of wrist movement. The only similarities of these two sports are they are racket sports and the both sports use the very similar muscle and arm movements.

2.4 Analysis of Badminton Game Activities

C. Z. Shan et al. [10] investigated dynamic data of upper limb movement including wrist, elbow and shoulder movement. They found that the wrist movement is the crucial parameter in badminton game. Amin Ahmadi et al. [11] investigated the translational and rotational motion of the swing using accelerometers for athlete skill assessment. The work of Chien-Lu Tsai et al. [12] analyze the muscular surface EMG activities pattern provides quantitative information about the exertion for the holding and swinging the badminton racket. This serve as an indication to the player of how much strength will be needed to execute the powerful smash.However, the work of [11] and [12] emphasized on bodily activities of the human in performing the action. The works of T. Jaitner et al. [13] in analyzing the racket speed provides good information about racket necessary for contributing a powerful shuttle ball speed. They only use accelerometers to correlate the ball velocity of the shuttle ball, lacked information about how hard the shuttle ball was being hit, and at which point of the racket hit the shuttlecock. The works of R. Jiang et al. [14] mentioned that the smashing performance is affected by stress and instantaneous speed of the upper arm based on the movement mechanics theory and law of conservation of energy, a standard or benchmark action for smashing can be regulate by using this quantitative model.

Therefore, it is visible that there have been numerous works of using motion sensors to recognize activity in different racket and based game. In case of badminton, multiple works have been done in the field of smash analysis and muscle movement during badminton smash. However, there have not been any significant work in recognition of other activities like service, smash return, forehand, backhand etc. Our work focus on recognizing these activities from accelerometer and gyroscope data. The output of this research will result in the recognition data of badminton activity. These data will be used in controlling a virtual reality (VR) badminton game. As controlling the game will require significant muscle movement, this game can be used as a medium of physical exercise. So it can be a solution to the lack of physical exercise scope. As physical exercise will be done through playing the game, people will be interested in participating in the game. So, it will be physical exercise medium for kids and will be a solution for obesity related issue.

Chapter 3

Methodology

We want to build a system where badminton game activity can be recognized using motion sensors. Later we can use this recognized activities in a virtual reality game that we want to implement. This virtual reality game will allow users to play badminton game indoor with real physical activity. So, it can be used as a medium of physical exercise for children and obese users. The MPU-6050 that we will be using, will give the acceleration and angular information. We will use these information and perform some operation to extract the features and this features will help us to recognize badminton activity and those will be used in the virtual reality game which is the main goal of this research. To achieve this goals, we need to integrate the parts of our system and perform operations with the system.

3.1 Hardware Configuration

Our main system will consists a sensor enabled badminton racket as the controller for users and a unity3D module for the virtual reality game. For recognizing activities we will use matlab simulation and perform recognition operation on the result. We have used an MPU-6050 sensor which contains an integrated accelerometer and gyroscope with the racket. This racket will be used as the input device in our system. That means user will make real badminton shot simulation with the bat. This move will be converted as an input in the virtual reality game. The sensor integrated bat looks like this:

Figure 3.1: Badminton Bat with Integrated Sensor

For reading the sensor values we have used an Arduino module (Arduino Mega). This arduino module reads the sensor values and send the data to our unity3D module. The unity3D receives the sensor data with some filtering then calculates the displacement in x,y,z axis. Then it uses this displacement values to move objects in the virtual reality game. We also created connection between the arduino module and Matlab module for our purpose of activity recognition. Because, In matlab, we are plotting the data for visualization so that we can understand the data and perform recognition operation.

3.2 System Architecture

We have designed our whole system as descibed in the system integration section. After the whole system was integrated the full system architecture looks like following.

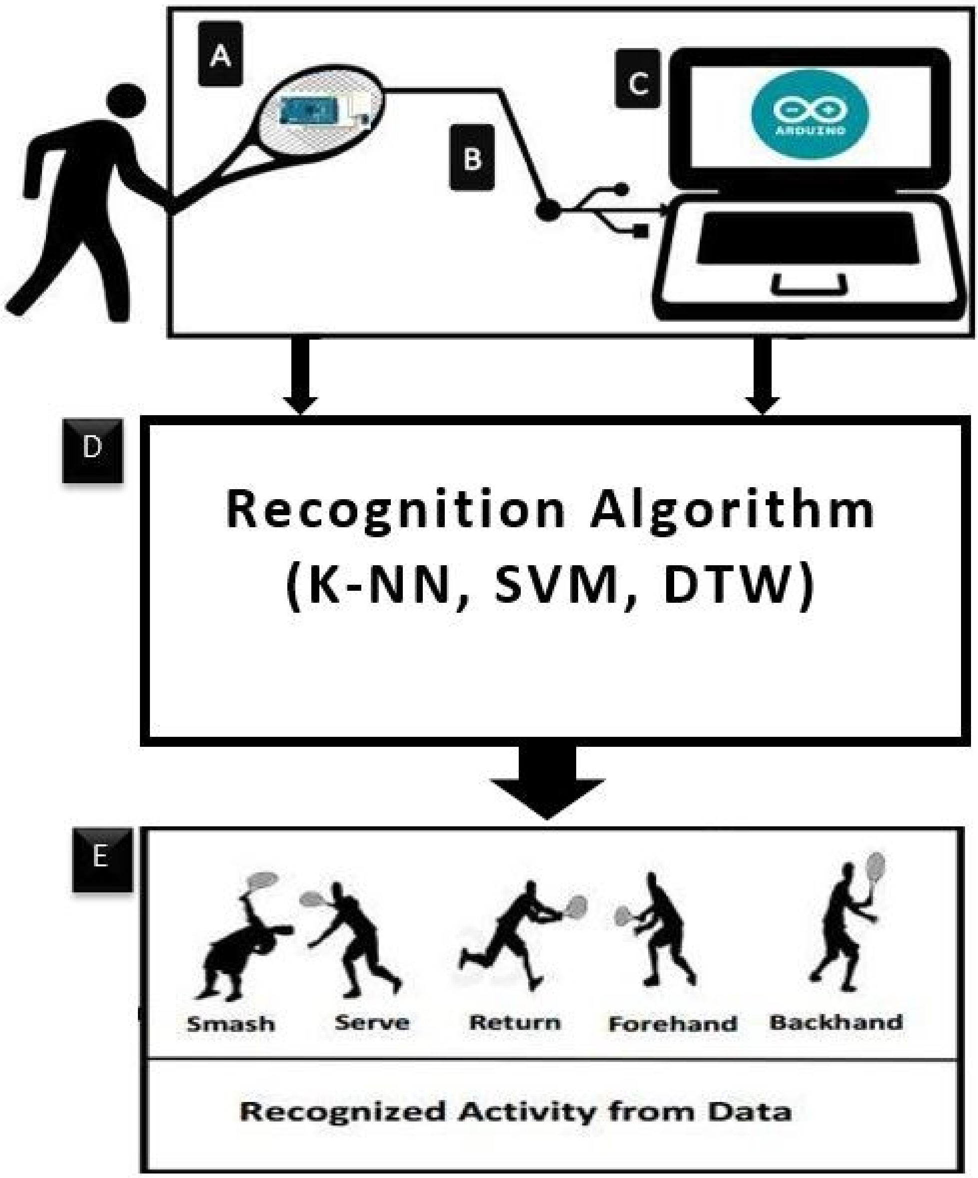

Figure 3.2: System Architecture of Badminton Activity recognition system

Here, In the system architecture diagram we see the important parts that make up the system. In figure 2(A) we see that a user will control the sensor enabled badminton bat. This sensor is embeded on a Arduino Mega board and it is connected to a computer via cable which we can see in 2(B). Now, we have used a wired arduino till now, but in our intended system we will be using a wireless arduino. In case of wireless arduino, we wil tranfer the data using a additional bluetooth module. Now, the sensor data is received by the arduino IDE via serial port communication. In 2(C) , it is explained that, the received sensor data will be processed by the arduino IDE. We will perform our filtering operation and calculate the displacement from the acceleration values in the arduino IDE. Now from the IDE we will send data to our recogtion module to recognize the activity performed by the user which is described in 2(D). The recogniton module will apply K-NN(K - Nearest Neighbor), SVM(Support Vector Machine), DTW (Dynamic Time Warping) or other machine learning algorithm to recognize the activities. In 2(E) we can see the recognized activities by the system. We can use this recognized activities to train AI(Arificial Intelligence) components and this trained AI can be used as opponents in virtual reality badminton game in future.

3.3 Defining Special Activities

We already defined that our main goal is to create a virtual reality game where user will control it using a sensor enabled badminton bat. User have to do real life shots simulation. So, controlling this game will be a medium of physical exercise. Now, for successful implementation of this virtual reality game, first we need to recognize the activities performed by the user. Because, If users just make random movements, that would mean nothing to the system. We have identified this following activities that are common in a real life badminton game.

i. Serve

ii. Return

iii. Smash

iv. Forehand

v. Backhand

So, we need to identify this activities performed by users at minimum. During each activity, we will get different sensors value. From this sensors value, we will recognize the activities performed by the user. It is to be noted that, more accurate recognition of activity will result in a good virtual reality game. If this activities are correctly recognized, then in the virtual reality game, we can make proper simulation of the action that the user is performing. So, users will be interested to play the game. Playing this game will help them with their regular exercise.

3.4 Data Collection

We will be using an MPU-6050 sensor to collect the required data for our activity recognition. MPU-6050 contains an accelerometer and a gyroscope sensor. It is very accurate as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it can capture the x, y, z channel at the same time. The sensor uses the I2C-bus to interface with Arduino. MPU-6050 provides us raw acceleration and gyroscope data in 3 coordinates. We collected the raw data and used it for our intended purposes.

We get the raw data from Arduino from its serial monitor. Later, we format the data according to our need. We also calculated the displacement in different axis from the accelerations. First we collected raw data from the arduino. Some snapshot of the data collection is given here:

```
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful
```

Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z
13252	-2760	8196	-307	-307	-307
13262	-2756	8180	-315	-315	-315
13246	-2750	8200	-317	-317	-317
13236	-2760	8190	-308	-308	-308
13246	-2772	8208	-314	-314	-314
13230	-2760	8210	-317	-317	-317
13246	-2770	8200	-310	-310	-310
13250	-2762	8178	-309	-309	-309
13246	-2752	8186	-317	-317	-317
13224	-2764	8190	-305	-305	-305
13242	-2756	8192	-309	-309	-309

Figure 3.3: Raw data of MPU-6050

Here,

Accel-X = Acceleration in X-axis.

Accel-Y = Acceleration in Y-axis.

Accel-Z = Acceleration in Z-axis.

Gyro-X = Gyroscope value in X-axis

Gyro-Y = Gyroscope value in Y-axis

Gyro-Z = Gyroscope value in Z-axis

3.5 Data Filtering and Preprocessing

Now we cannot really understand the data clearly. Because it is not in a normal acceleration or angle format. To understand the data and make it usable first we need to convert the data in a regular format and apply some filtering to reduce the noise and the error in data. The MPU-6050 has a digital low pass filter (DLPF) for both the gyroscope and accelerometer. Which means that the low pass filter only allows lower frequencies to pass and filters out higher frequencies such that come from the sensor.

Mpu-6050 also comes with a DMP which means Digital Motion Processor. It is also called a "Digital Motion Processing Unit". This DMP can be programmed with firmware and is able to do complex calculations with the sensor values [15]. DMP fuses the accelerometer and gyroscope data together to minimize the effects of errors inherent in each sensor [16]. It is to be noted that, we have not applied any additional noise filter. But, in our future works, we can use Kalman filter, Complementary filter if we need.Though we did not use any external noise reduction filter, we still had to convert the raw data to a readable format. To convert the raw acceleration and gyroscope value to an understandable format, we used the help of a library built by Omer Ekram Ul Haq named MPU6050_Arduino [17]. In the library, first the data was calibrated. We calculated an offset for the starting position and then using that offset, we calculated the acceleration and angle. Here is a snapshot of the calibration and offset calculation.

```
Resetting MPU6050 and waking it up.....
The gyro scale is set to 7.63 milli Degree/s
The accel scale is set to 1.20 milli m/s^2
Calibrating gyroscope .... dont move the hardware ..........
.............................................................
gyro_x register offset = -272
gyro_y register offect = 65
gyro_z register offset = 78
Calibrating accelrometer .... dont move the hardware ..........
.............................................................
Accel_x register offset = -620
Accel_y register offect = 8197
Accel_z register offset = -953
```

Figure 3.4: Setting up the offset for MPU-6050

After calculating the offset, we calculate the acceleration and angle of gyroscope in regular format considering the offset. Below, there is a snapshot showing the formatted data.

Accel-X	Accel_Y	Accel-Z	Angle-X gyro	Angle-Y gyro	Angle-Z gyro	Angle-X Accel	Angle-Y Accel	Angle-Z Accel
0.15	-7.01	10.38	-0.15	0.21	0.24	0.84	29.23	55.98
0.10	0.03	9.74	-0.22	1.55	2.19	0.57	-0.18	89.41
0.08	0.04	9.94	-0.60	1.62	6.72	0.44	-0.26	89.50
0.19	0.04	9.91	-1.08	1.25	11.13	1.09	-0.23	88.90
0.22	0.05	10.06	-1.41	1.36	14.87	1.23	-0.29	88.75
0.19	0.05	9.88	-1.30	1.35	18.90	1.08	-0.30	88.89
0.25	0.04	9.96	-0.84	1.69	22.12	1.41	-0.23	88.58
0.22	0.04	9.95	-1.16	1.22	25.42	1.29	-0.24	88.70
0.14	0.03	9.93	-1.36	1.21	28.70	0.81	-0.18	89.18
0.19	0.05	9.91	-1.50	1.27	31.92	1.09	-0.31	88.88
0.15	0.03	9.87	-1.06	1.46	34.19	0.86	-0.16	89.13
0.20	0.04	10.00	-0.85	1.95	37.20	1.13	-0.21	88.86
0.15	0.03	9.95	-0.87	2.09	39.56	0.87	-0.19	89.12
0.17	0.03	9.93	-0.77	2.31	42.27	0.99	-0.18	89.00
0.17	0.03	9.91	-1.14	2.19	44.59	1.00	-0.17	89.00

Figure 3.5: Formatted MPU-6050 data

We will use this formatted data to extract the feature and recognize activity in future.

3.6 Feature Selection

After processing the data we got acceleration and gyroscope values in each of the three axis. Now, in the goal to recognize badminton game activity correctly, we need to identify the features that are most important for the recognition based on our dataset. We have tried out different combinations and found out the required number of features for good recognition rate. The selected feature list is given below with small description.

Feature	Description
Mean	The average value of the signal over the window
Median	Median signal value over the window
Standard Deviation	Measures the spreadness of the signal over the window
Variance	Square value of standard deviation
RMS	Root mean square value of the signal
Highest Peak	Positive peak value of signal
Lowest Peak	Negative peak of signal
Energy	Calculated by applying FFT on the signal and taking absolute value

Table 3.1: Features with small description

We have calculated all this feature values separately for the three accelerometer axis values and three gyroscope axis values. Each of the selected features are important for the recognition process. Each of them helps to differentiate one activity from others. For example, Mean, Median, Standard deviations values are common features that are selected. We also selected the variance which is the squared values of standard deviation. When the deviation between two different activities is small, then the variance value will help us to differentiate. Similarly, other features are selected because they bear significant importance in recognition of the badminton activities.

After selecting the features we created feature vector for each instance of every activity. As we have calculated 8 features for each of the accelerometer axis and each of the gyroscope axis, we have a total of (6 x 8) = 48 features in each of the feature vectors. The feature vector looks like the following.

< m_ax, med_ax, sd_ax, var_ax, rms_ax, hp_ax, lp_ax, energy_ax >

Here

m_ax = Mean value of acceleration,

med_ax = Median value of acceleration,

sd_ax = Standard deviation,

var_ax = variance of acceleration,

rms_ax = Root mean square value acceleration,

hp_ax = Highest peak of acceleration,

lp_ax = lowest peak of acceleration,

energy_ax = Energy value of acceleration, in the x-axis.

Chapter 4

Recognition of Activities

We have integrated the whole system and make it ready for the test run. In our system, users will perform badminton game activity using the sensor enabled racket. From the sensors data, we will try to recognize activities. We selected features from the filtered sensor data. After selecting features from our dataset, we moved on to recognize the badminton games activity. For this we performed different type of experiment based on different algorithms. and got results varying from good to bad. Some experiments were performed on the features and some were performed on the filtered raw data.

4.1 Fixing the Axes and Initial Positioning

Before we started to take data from the MPU-6050, we have to calibrate the accelerometer and he gyroscope to fix its axis. Once its axis is fixed, the MPU automatically adjust the axis according to the movement Add Referenece. So, we dont need to consider the axis changing issue during movement. For making the calculation process simpler, we have set an initial position before performing any activity. When not playing a shot, it is intuitive for the player to fix the racket in a position which normally vary from player to player. For simplicity, we have fixed the same position for every player. Also, a shot can be played using different techniques. But in our experiment we have considered only one way of making a badminton shot for simplicity.

4.2 Calculating Root Mean Square (RMS) value

RMS value of a set of values is the square root of the arithmetic mean of the squares of the values. In case of a set of n values (x1,x2,x3,.,xn), the rms value will be

$$x_{rms} = \sqrt{\frac{1}{n}\left(x_1{}^2 + x_2{}^2 + \cdots + x_n{}^2\right)} \tag{4.1}$$

To calculate the combined signal of the x,y,z axis, we decided to calculate the rms value for every set of acceleration. We followed the given formula to calculate the rms value for our defined activities. For example, we calculated the rms value for three different activity (serve, smash, and backhand) for a user. We plotted the data in matlab to depict the picture. We have only considered the Smash, Serve, Backhand activity in this experiment and on other feature based approaches.

For serve activity,

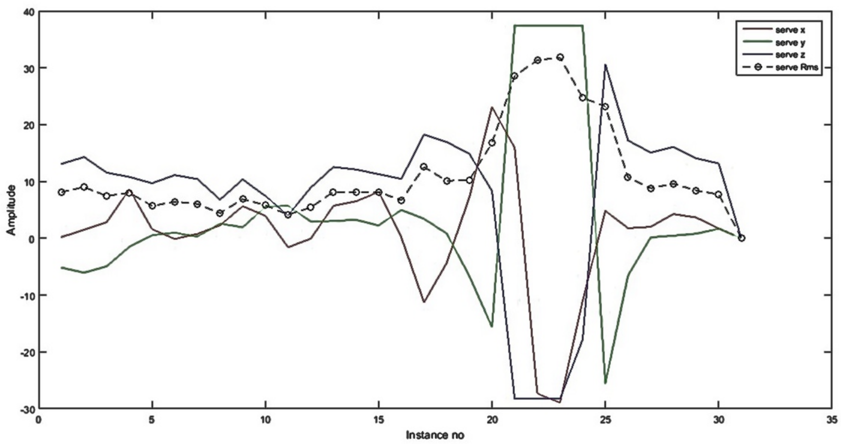

Figure 4.1: Serve activity acceleration comparison

For smash activity,

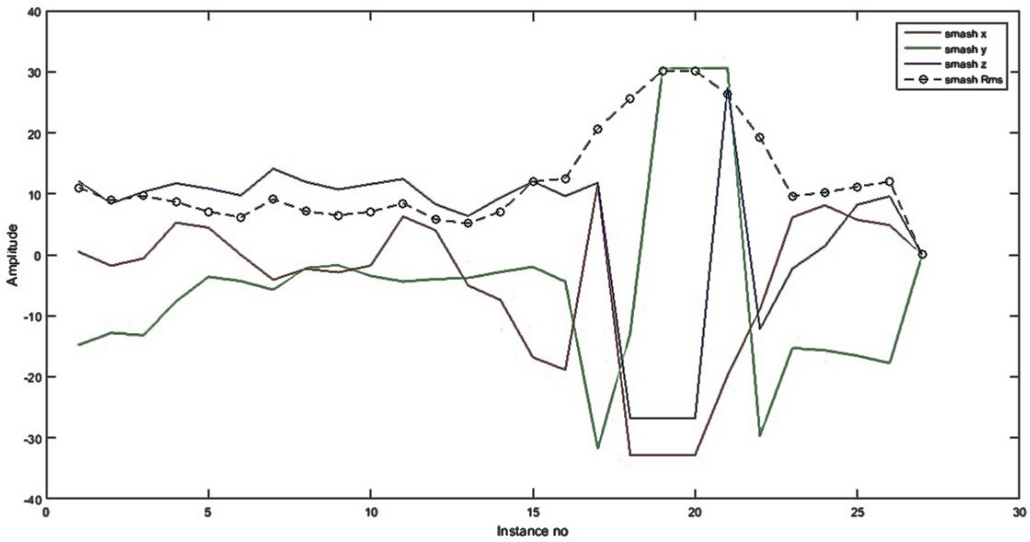

Figure 4.2: Smash activity acceleration comparison

For backhand activity,

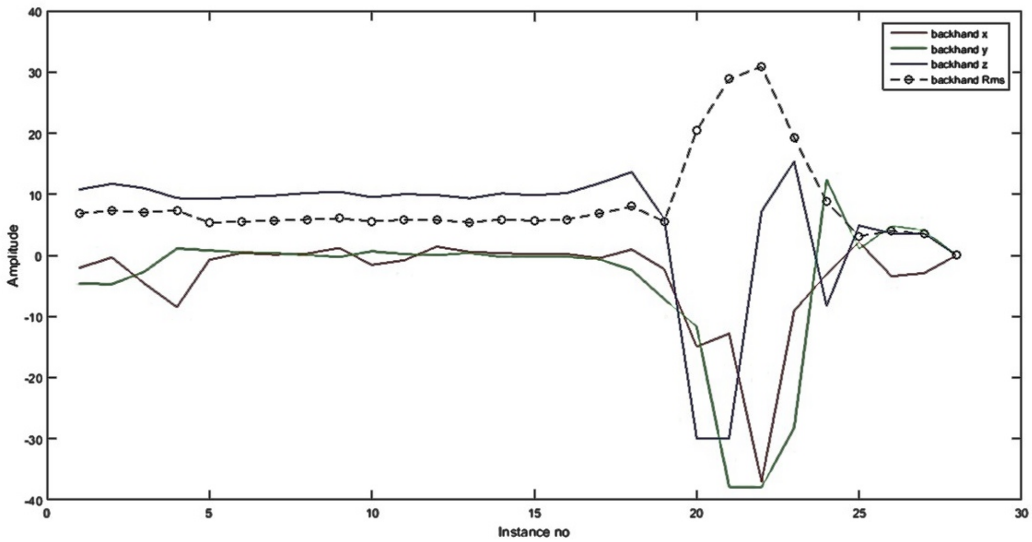

Figure 4.3: Backhand activity acceleration comparison

We analyzed the result that we got by calculating the rms and plotting into matlab. The detail result is discussed in the result analysis section.

4.3 Applying K-Nearest Neighbors Algorithm (k-NN)

K-Nearest Neighbors (k-NN) is a offline classification algorithm. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors where k is a positive integer number. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. k-NN has previously been used in sensor based research work [18] [19]. We have applied k-NN on our dataset. We have created training feature vectors from the training datasets and compared our test feature vectors with them. We have used done the experiment for different values of k (k=1,2,3..) and compared the result.

4.4 Applying Support Vector Machines(SVM)

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In SVM, we give a set of training examples and the classifier build a model that assigns new examples into one category. So, it works as a non-probabilistic binary linear classifier. When the data are not labeled, supervised learning is not possible. So labeled data is a requirement for SVM. SVM has been used previously in different types of activity recognition. It has been used to recognize human physical activities [20] and also in many other different kinds of activity recognition [5] [21]. We have applied SVM on our dataset. We have created two sets of data from our dataset. The bigger set is used as the training set and the smaller set is used as the test set.

4.5 Applying Dynamic Time Warping (DTW)

Dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two given (time- dependent) sequences under certain restrictions. It aims at aligning two sequences of feature vectors by warping the time axis iteratively until an optimal match (according to a suitable metrics) between the two sequences is found [22].

Consider two sequences of feature vectors:

A= a1, a2, a3..aN

B= b1, b2, b3.bN

The DTW algorithm optimizes the match in the following manner:

P = min (d (A, B))

Here d = distance between A and B

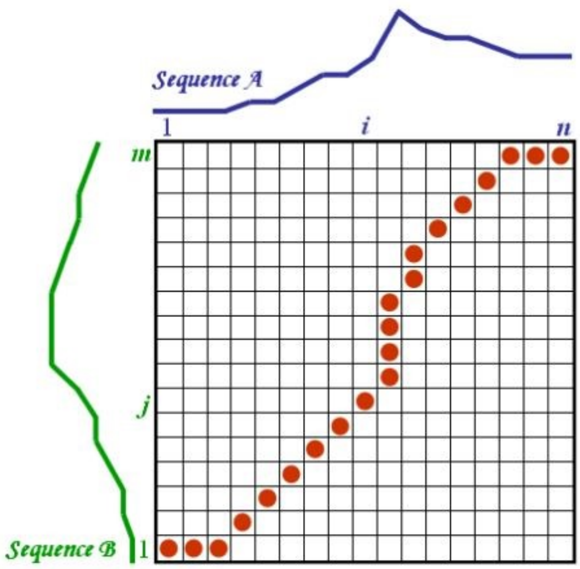

Figure 4.4: Grid view of DTW algorithm

To find the best match or alignment between these two sequences one need to find a path through the grid which minimizes the total distance between them. The procedure for computing this overall distance involves finding all possible routes through the grid and for each one compute the overall distance.

DTW has been used previously in recognizing human activities [23]. In DTW algorithm, we compare two sequence of time dependent signal and find the distance between those sequences. In our case of activity recognition in a badminton game first we select some template sequences for each activity. We selected the template sequences from our dataset randomly. We will use this template to compare the other time dependent sequences and the find the distance between a test sequence and all the template sequences. After this we will compare all the dtw distance found from the comparison. We will label the test sequence with the label of the least distanced template.

Chapter 5

Result Analysis

We performed different type of experiment based on different algorithms and got results varying from good to bad. Some experiments were performed on the features and some were performed on the filtered raw data. Here we will be explaining the results in details.

5.1 Results from Root Mean Square (RMS) value

From the graph of RMS values of different activities, it is quite evident that, for every activities RMS value are similar. It is due to different reasons. Firstly, though each activity is different from one another, the pattern that users follow when performing activities are similar. Because when performing activities the acceleration mainly changes during a certain period.

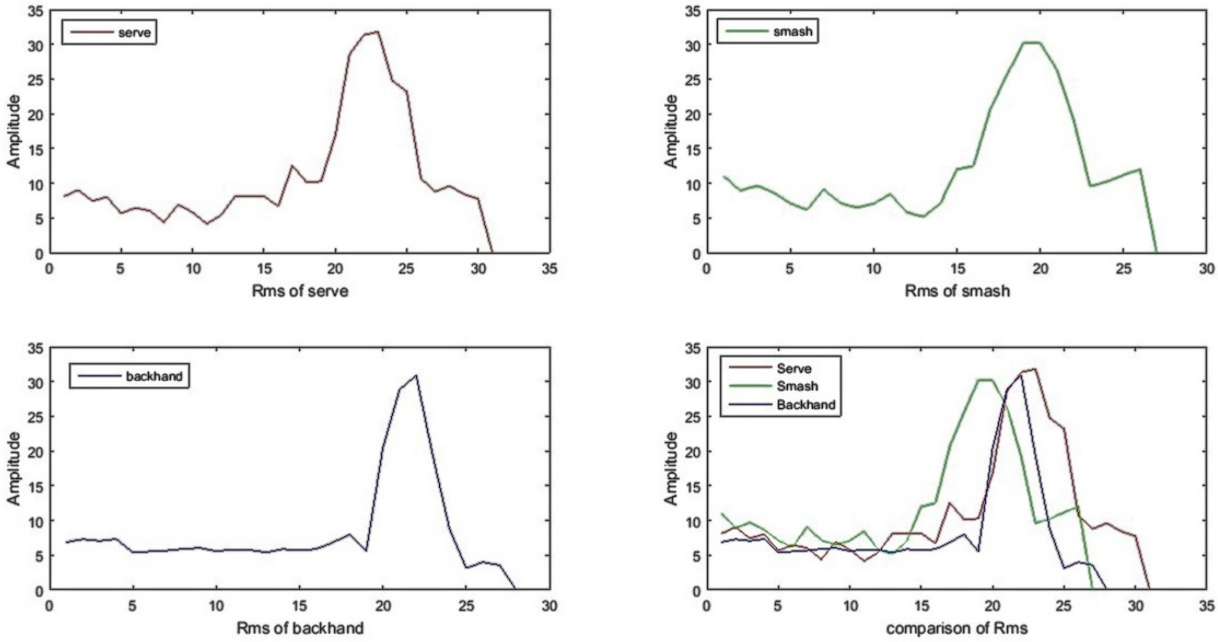

Figure 5.1: RMS value comparison

The difference is that, the changes are in different axes for different activities. But when we are calculating the RMS, we are actually ignoring separate axis and combining all axes to generate a single value. So, even if the changes were in different axes, the RMS value of different activities becomes almost the same. Also, for applying the RMS, the values are first squared and then root value is calculated. For this, the peak in negative axis is also moving to positive axis. The peak of RMS value will always be positive. That's why we have come to the decision that, considering the RMS value for comparison wouldn't yield good result.

5.2 Results from K-Means Clustering

We also tried to apply k-means clustering on our dataset. But, as our activities are discrete events, so it is not possible to have clusters of different activities data in a particular dataset. For example, when we are applying k-means clustering on a smash activity data, it is never possible that we will have clusters of other activities data in that dataset. There may be some similarities in some portions, but the dataset will always be different. Thats why we wont get any significant result by applying K-mean clustering in our case.

5.3 Results from K-Nearest Neighbors (k-NN)

K-NN classification assigns a class to each of the test data based on the training data. So, at first we chose some feature vectors from our dataset as the training dataset and the rest as the test dataset. Then, we compared the test feature vectors to the training feature vectors of activities using k-NN. Using k-NN classification in our dataset, all the test data was assigned to a training activity class. We have used three activities (smash, serve, and backhand) dataset for testing. We have used different values of k and found out that k=1 is better suited in our case. After analyzing the result we found out that, the classification gives us around 58% accurate recognition of activities on average.

5.4 Results from Support Vector Machines (SVM)

We have prepared our dataset containing three activities data. We have used the data of smash, serve and backhand activity. We had a dataset of 180 instances. We separated the dataset into two portions. 80% data was used as the training data and rest 20% was used as the test data. We separated the data using Resample filter in Weka machine learning tool [24].We applied SVM on the test data using the Weka machine learning tool [24] and got an 88.89% accuracy on the test data. We have generated a confusion matrix table that shows us the details of the classifications. In the confusion matrix in the following table, each cell shows us the no. of instances for a given class (row labels show the true class, and column labels show the predicted class). Correct predictions are given on the diagonal cells. We can see that, among these three activities, backhand can be recognized with more accuracy and the accuracy of serve recognition is low.

	Smash	Serve	Backhand	Accuracy
Smash	**12**	**0**	**0**	**100%**
Serve	**1**	**9**	**2**	**75%**
Backhand	**1**	**0**	**11**	**91.67%**

Table 5.1: Confusion Matrix Table for SVM

$$\textbf{Overall accuracy} = \textbf{(Correctly identified instances)/(Total number of instances)}$$

$$= \mathbf{88.89\%}$$

5.5 Results from Dynamic Time Warping (DTW)

In DTW algorithm, we compare two sequence of time dependent signal and find the distance between those sequences. It is one kind of online classifier. In our case of activity recognition in a badminton game first we select some template sequences for each activity. We selected the template sequences from our dataset randomly. We will use this template to compare the other time dependent sequences and the find the distance between a test sequence and all the template sequences. After this we will compare all the dtw distance found from the comparison. We will label the test sequence with the label of the least distanced template.

Now we had a total of 5 activities. We have taken 6 template sequence for each of the activities. So, if we consider all the activities in this experiment, then in total we have 30 template sequence. Now we will compare our rest of the sequence of the dataset with these templates. We calculated the dtw distances between the test sequences and the template sequences using the algorithm. Then we labeled the test sequences based on the least distance. We generated the confusion matrix to show the recognition details.

	Smash	Serve	Backhand	Forehand	Return	Accuracy
Smash	**19**	**4**	**1**	**2**	**7**	**57.57%**
Serve	**1**	**16**	**1**	**7**	**8**	**48.48%**
Backhand	**1**	**8**	**14**	**6**	**4**	**42.42%**
Forehand	**4**	**9**	**0**	**14**	**6**	**42.42%**
Return	**0**	**9**	**1**	**3**	**20**	**60.60%**

Table 5.2: Confusion Matrix Table for five activities using DTW

Overall accuracy = (Correctly identified instances)/(Total number of instances)

= 83/165 = 50.30%

From the result we can see that the accuracy rate is pretty low. By analyzing the misclassification result we can see that the misclassification rate is higher among 3 particular activities. The activities are Serve, Smash and Forehand. To find out the reason we analyzed these activities to little details. By doing this we came to a decision which can explain the misclassification reasons. In below, we can see some pictures representing the activities.

Figure 5.2: Comparison of Serve, Smash and Forehand

From these pictures we can see that there is some similarities among all these activities. If we compare serve and forehand, we can see that except the starting point of these activities, both the activities have a similar sequence of simulation of the hand. Again, serve and return is almost similar in every aspect. So, it can also be said via inference that there is similarities between forehand and return. With this three activities having similarity among themselves, the template of this activities are also similar. So, when we calculate the distance between test and template sequences, all these three activities gives distance which are similar. So, we do not get the correct label all the time. As we are classifying based on the least distance, so the classifier analyze the distances and label the activity based on the distance. As we can get the least distance from any of these three activities due to their similarity, the misclassification rate increases with it.

As we need better accuracy rate, we tried to solve this misclassification problem. To do that, first we increased the number of template sequences. We thought if we increased the number of template sequences, then the probability of getting the least distance from the actual activity will increase. So, that will help to better the accuracy. But, that is not the case on every attempt. It is not guaranteed that increased number of template sequence will result in better accuracy. Because, our new templates can also be misleading if they have less distance to other class activities rather than the templates of its own class. We increased our template sequences to 18 instances per activity. So, now we have a total of 90 template sequences. We compared the rest of the sequences as test sequences with all the template sequence and we got the following result shown in the confusion matrix.

	Smash	Serve	Backhand	Forehand	Return	Accuracy
Smash	**13**	**4**	**1**	**2**	**1**	**61.90%**
Serve	**0**	**12**	**1**	**4**	**4**	**57.14%**
Backhand	**3**	**5**	**10**	**1**	**2**	**47.61%**
Forehand	**1**	**4**	**2**	**11**	**3**	**52.38%**
Return	**0**	**9**	**0**	**3**	**9**	**42.85%**

Table 5.3: Confusion Matrix Table for five activities using DTW

$$\textbf{Overall accuracy} = \textbf{(Correctly identified instances)/(Total number of instances)}$$

$$= \mathbf{55/105 = 52.38\%}$$

So, we can see that after tripling our template sequences we can only improve our accuracy by only 2%. We have already explained the reason behind it. So, increasing the number of templates in not a good approach to improve recognition accuracy.

Previously, when we classified our dataset with offline classifiers we only considered three activities for simplicity. The activities were Smash, Serve and Backhand. We extracted many features from our datasets and created feature vectors for each of the activity. Then we divided our whole dataset in two sets randomly (80% for training set and 20% for test set). Then, we applied SVM classifier on our dataset and got around 90% accuracy. We also applied random forest algorithm on the dataset and got around 68% accuracy. So, we tried to figure out the reason behind it and found it out. It appears that, these three activities are actually pretty different considering the three similar activities (Serve, Forehand and Return). And as we have run the classification algorithms on activities with significant differences, we had got better accuracy.

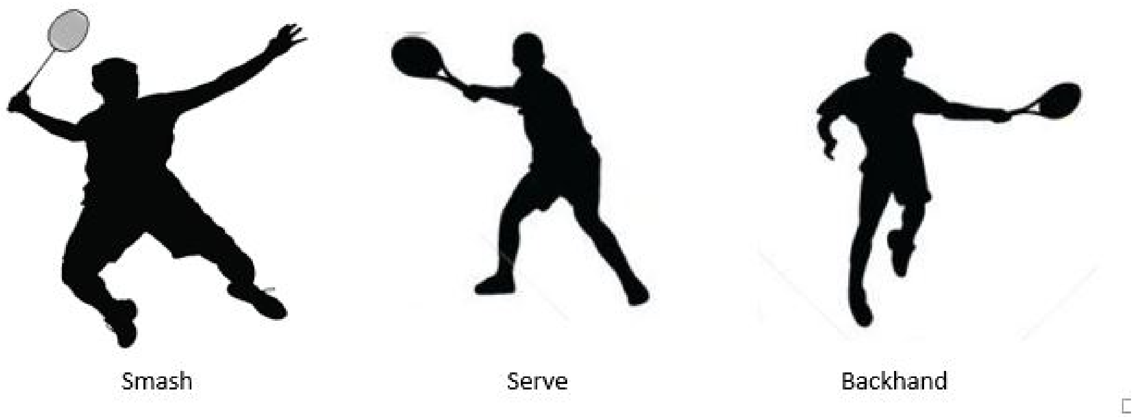

Figure 5.3: Comparison of Smash, Serve and Backhand

So, we decided to run DTW only on these three activities (Smash, Serve and Backhand). Two instance of applying DTW on these three activity is given below.

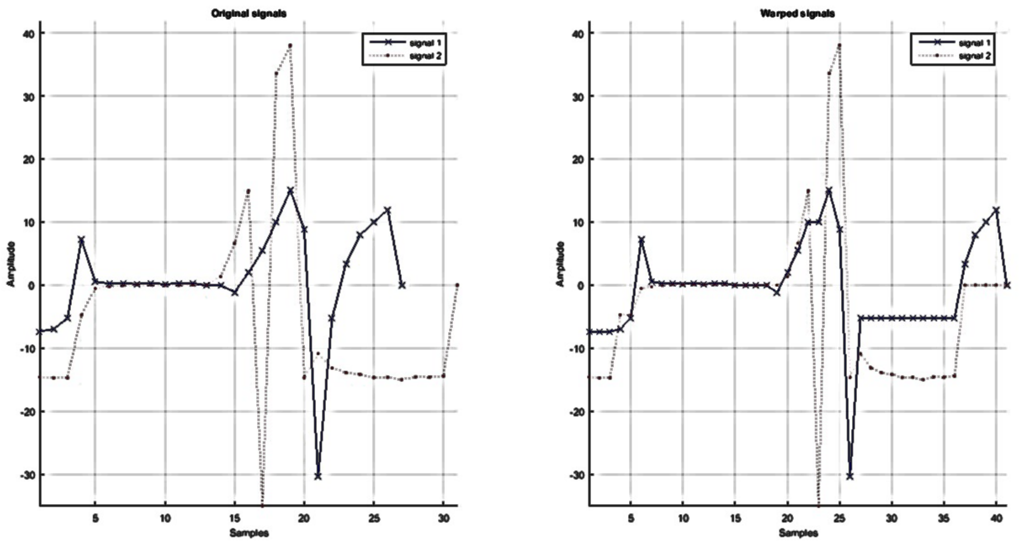

Figure 5.4: DTW on a Smash (test) and Serve (reference)

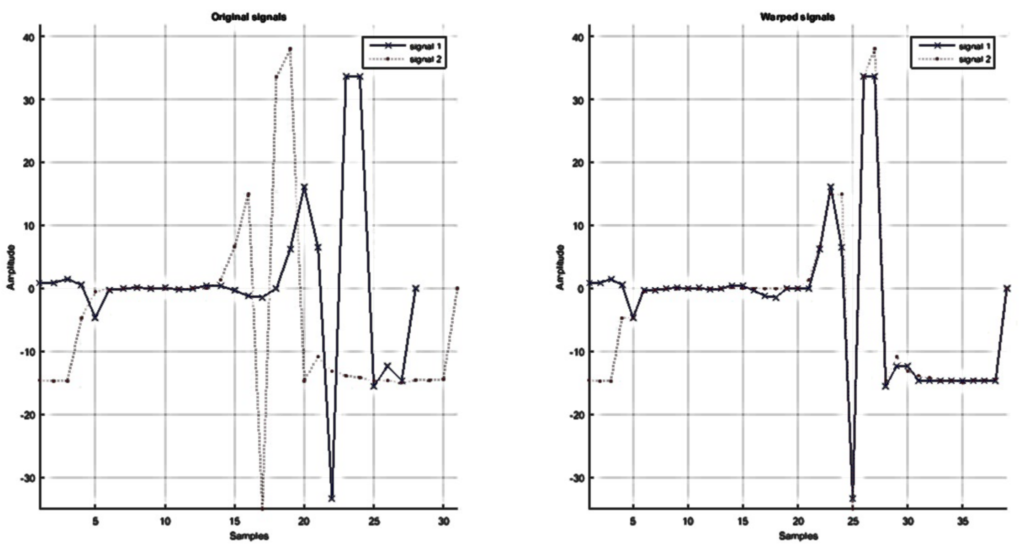

Figure 5.5: DTW on a Smash (test) and Smash (Reference)

Here, in the first figure we can see that, DTW is warping a test return signal to a reference smash signal. In the figure, the dark line is the test signal and the blurred line is the reference signal. DTW is warping the test signal to the reference signal. We can see that, even after warping, the signals shows some dissimilarity. It is because the reference is a smash activitys signal and test signal is from return activity. But, in the second figure, we took a smash activitys signal as the test signal. So, after warping the signals become quite similar. So, in this case, the dtw distance is less than the previous case.

Now, considering the specified three activities (Smash, Serve, and Return) we got better result comparing the previous results. The result that we got is given below in the confusion matrix.

	Smash	Serve	Backhand	Accuracy
Smash	**27**	**4**	**2**	**81.81%**
Serve	**3**	**28**	**2**	**84.84%**
Backhand	**3**	**5**	**25**	**75.75%**

Table 5.4: Confusion Matrix Table for three activities using DTW

Overall accuracy = (Correctly identified instances)/(Total number of instances)

= 80/99 = 80.81%

So we can see that for these three activities with significant differences the algorithm can correctly classify 81% activities. But if we take consideration of each of the five activities, then the misclassification rate increases and with that the percentage of correctly identified activities decreases.

Chapter 6

Future Works and Conclusions

One of our biggest challenge of the future is finding out a proper way to find out how to accurately recognize all the five activities mentioned in this research. Till now we have only figured out some stable ways to recognize only three activities with significant difference. When we are dealing with all five activities all at once, the recognition rate falls down as some of the activities are way too similar. So, we need to find out the perfect feature for classification or any better suited algorithm to increase the recognition rate in case of five activities. We also may need to tinker our physical device settings. We thought of adding an extra sensor on the humans body to get extra movement of body during performing activities. This may give us the extra feature when dealing with similar activity. We thought of these approach at the very last moment. So, we could not complete this phase in this scope. This possess us with greater challenge in the future.

After analyzing the results from k-NN,SVM and DTW, it is quite obvious to say that the SVM classifiers gives us better results in recognizing the activities. But, one important thing to note is that k-NN, SVM, k-Means clustering etc. are offline classifiers used and DTW was an online classifier used to classify our activities. We can try to increase the stability of the accuracy by further research and with this improved accuracy, the recognized activities can further be used to train Artificial Intelligence(AI) based applications such as various virtual reality games, where we can control the difficulty level of the AI component with much more precision.

Appendix A

Configuring Arduino and MPU-6050

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on the computer, used to write and upload computer code to the physical board.

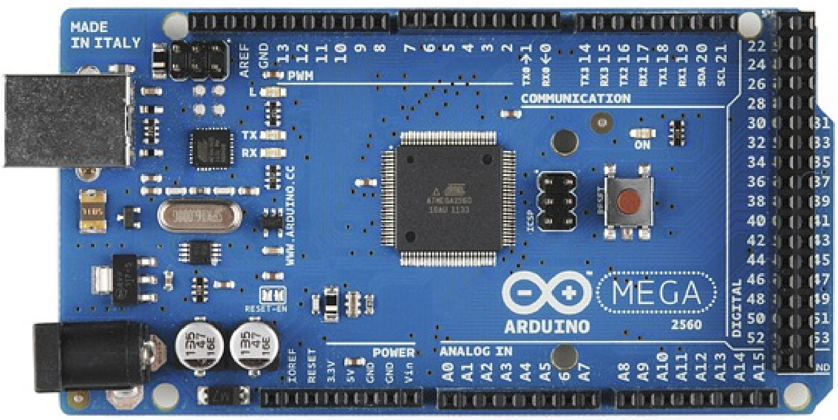

Figure A.1: Arduino Mega-2560

Now, MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefor it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyroscope.

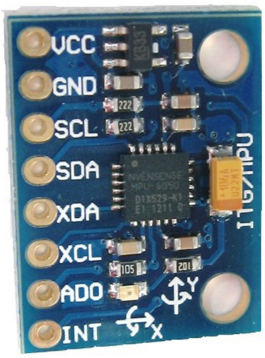

Figure A.2: MPU - 6050

The MPU 6050 is a 6 DOF (Degrees of Freedom) or a six axis IMU sensor, which means that it gives six values as output. Three values from the accelerometer and three from the gyroscope. The MPU 6050 is a sensor based on MEMS (Micro Electro Mechanical Systems) technology.

The MPU 6050 communicates with the Arduino through the I2C protocol. The MPU 6050 is connected to Arduino as shown in the following diagram. Here, if your MPU 6050 module has a 5V pin, then we can connect it to your arduinos 5V pin. Else, we will have to connect it to the 3.3V pin. Next, the GND of the arduino is connected to the GND of the MPU 6050. Then we connect arduinos digital pin 2 (interrupt pin 0) to the pin labelled as INT on the MPU 6050. Next, we need to set up the I2C lines. For this connect the pin labelled as SDA on the MPU 6050 to the arduinos communication pin SDA. And the pin labelled as SCL on the MPU 6050 to the arduinos SCL pin. And this way we have finished wiring up the Arduino to the MPU 6050.

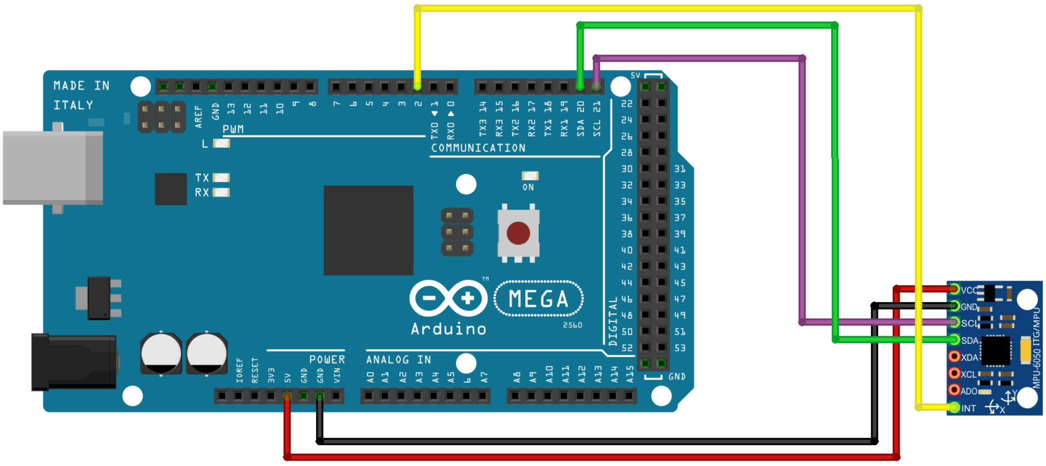

Figure A.3: Interfacing Arduino with MPU-6050

Appendix B

Data Collection steps

After interfacing the Arduino with the MPU 6050, we integrated the Arduino module with our badminton bat. We used extension cable to connect the sensor enabled bat to the computer. Then we begin our data collection steps. We picked different set of users for taking data of the defined activities. Our users varies in age, sex, body types. This varied type of users helps us to create a dataset that consists data of different types of user. We instructed the users to perform the five defined activities. For some activities we restricted the user to perform the activity in a certain way. For example, A serve can be done in two different ways. But we restricted all of our users to perform the serve in a same way. Similarly, we have done the same thing for all the defined activity. This is done to keep the simplicity. If a particular activity is done in different ways then it is almost impossible for the algorithms to recognize different ways of activity as a single activity. So, to keep the recognition procedure simple, we specified a particular way to perform each activity.

Every users performed their activities in a specified way. We collected the data from 20 users. Each user performed three instances per activity. That means, for a single user, we have (3 x 5) = 15 instances of data where each activity have 3 instance of data.We tested the defined activities with several types of users.The users varied in age, height and weight. This gave the diversity that was significant in correctly identifying or more precisely recognizing the activities.In the following table we can see the information about those users upon whom the activities were tested.

Table B.1: Set of Users in Data Collection

Serial	**Age**	**Height (cm)**	**Weight(Kg)**
1	21	176	68
2	23	180	70
3	31	168	61
4	21	175	75
5	22	175	81
6	22	172	62

From the table we can see that, the users in our dataset varies in height and width. We tried to maintain this variety because this would give more stable recognition from our system. It is due to the fact that the real users of the system will also vary in height and width.

We stored the real time data generated by performing different activities in separate .csv (Comma Separated Values) file. So, each instance of each activity were stored in separate files. Each files contains 6 columns of data. These columns contain data of three accelerometer axes and three gyroscope axes. Data was taken from when a user starts performing an activity to the finish of that activity. These way our whole dataset was generated and were used later on.

Bibliography

[1] D. E. Warburton, C. W. Nicol, and S. S. Bredin, "Health benefits of physical activity: the evidence," *Canadian medical association journal*, vol. 174, no. 6, pp. 801–809, 2006.

[2] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.

[3] D. C. Eunju Kim, Sumi Helal, "Human activity recognition and pattern discovery," in *IEEE Pervasive Computing, 2009*, 2009.

[4] D. S. Attila Reiss, Gustaf Hendeby, "A competitive approach for human activity recognition on smartphones," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, pp. 455–460, ESANN, 2013.

[5] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *AAAI*, vol. 5, pp. 1541–1546, 2005.

[6] M. D. F. C. L. O. Ferhat Attal, Samer Mohammed and Y. Amirat, "Physical human activity recognition using wearable sensors," in *Sensors*, 2015.

[7] O. T. Kerem Altun, Billur Barshan, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," in *Pattern Recognition*, pp. 3605–3620, 2010.

[8] M. Lapinski, E. Berkson, T. Gill, M. Reinold, and J. A. Paradiso, "A distributed wearable, wireless sensor system for evaluating professional baseball pitchers and batters," in *2009 International Symposium on Wearable Computers*, pp. 131–138, IEEE, 2009.

[9] "ZEPP TENNIS," ZEPP US Inc. [Online]. Available: http://www.zepp.com/tennis/ [Last Accessed On: 23 May 2016, 09.51 PM].

[10] C. Z. Shan, E. S. L. Ming, H. A. Rahman, and Y. C. Fai, "Investigation of upper limb movement during badminton smash," in *Control Conference (ASCC), 2015 10th Asian*, pp. 1–6, IEEE, 2015.

[11] A. Ahmadi, D. D. Rowlands, and D. A. James, "Investigating the translational and rotational motion of the swing using accelerometers for athlete skill assessment," in *2006 5th IEEE Conference on Sensors*, pp. 980–983, IEEE, 2006.

[12] C.-L. Tsai, C.-C. Yang, M.-S. Lin, and K.-S. Huang, "The surface emg activity analysis between badminton smash and jump smash," in *ISBS-Conference Proceedings Archive*, vol. 1, 2008.

[13] T. Jaitner and W. Gawin, "Analysis of badminton smash with a mobile measure device based on accelerometry," in *ISBS-Conference Proceedings Archive*, vol. 1, 2007.

[14] R. Jiang and Z. Wang, "Analysis on smashing motion in badminton," in *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, pp. 651–657, Springer, 2013.

[15] DMP: Digital Motion Processing. [Online]. Available:http://playground.arduino.cc/Main/MPU-6050/ [Last Accessed On: 23 May 2016, 10.08 PM].

[16] DMP: Digital Motion Processing. [Online]. Available:http://www.geekmomprojects.com/mpu-6050-dmp-data-from-i2cdevlib/ [Last Accessed On: 23 May 2016, 10.01 PM].

[17] MPU-6050 and Arduino Calibration. [Online]. Available:https://github.com/helscream/MPU6050-Arduino [Last Accessed On: 23 May 2016, 10.26 PM].

[18] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pp. 4–pp, IEEE, 2006.

[19] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[20] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *International Conference on Ubiquitous Intelligence and Computing*, pp. 548–562, Springer, 2010.

[21] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete consine transform and svm," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pp. 5041–5044, IEEE, 2009.

[22] DTW: Dynamic Time Warping. [Online]. Available:http://web.science.mq.edu.au/ cassidy/comp449/html/ch11s02.html [Last Accessed On: 05 November 2016, 11.25 PM].

[23] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pp. 1–5, IEEE, 2011.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.