Unlocking Wearable Device by Body Movement

Submitted by:
Shouwei Chen,
Qingxiao Zhao,
Xiaopei Zhang

Team Project Number: 5

Advisor:
Yanyong Zhang

April 29, 2015

Submitted in partial fulfillment of the requirements for senior design project

Electrical and Computer Engineering Department
Rutgers University, Piscataway, NJ 08854

Abstract

Some wearable devices are not easy to be unlocked by input the password. This time we would like to find a way to unlock MOTO 360 by movement of hand. We use different algorithms to recognize the movement pattern and compare the performance of Dynamic Time Warping (DTW), fast DTW and Hidden Markov model (HMM).

Table of Contents

1. Introduction

1.1. Background information

Intelligent devices are important parts of our life. Not only the smart phone, but also the wearable device has become an everyday topic and is making its entry into our lives. Wearable devices are clothing and accessories with appropriative computer and advanced electronic and computer technologies. In 1960s, MIT Media Lab presented this innovative technology, which can be put multimedia, sensors and wireless communication technology embedded in people's clothing, gestures and eye movements can support multiple operation species interaction. Wearable technology is an important part of ubiquitous computing and the history and development of wearable computers. Ubiquitous computing and wearable technology share the vision of interweaving technology into the everyday life. The history of wearable technology is influenced by both of these responses to the vision of ubiquitous computing. With the internal connectivity of wearable devices, it is possible to make fast data acquisition, holding social connections through ultra-fast ability to efficiently share content. We can have seamless Internet experience without the traditional handheld devices. Such like Google Glass, Fitbit, Apple Watch, and MOTO 360. With all kinds of practical functions, these stuffs help us to keep in touch with new information and advanced technologies in daily life.

Though wearable devices supplies an advantage approach for keep in touch with smart devices for us, we still cannot ignore a functional defect of them: for most of wearable devices are not easy to be unlocked by input the password, because of small screen and lack of buttons. But we can find that wearable still have some sensors, like acceleration transducer, gyroscope, and so on. The sensors still provide the developers with a wide variety of possibilities. Using the data from acceleration transducer to record the movement of our hand and unlocking the devices by these data is a good try.

1.2. Goals

We want to design an app for unlocking the MOTO 360, which is one kind of wearable device with android system. We need to ask the user to do a series of hand movement, and then compare it with the pre-set one. So we should build a system, which can:

1. Ask the user to set a hand movement as the standard movement, just like we set a password for our cell phone. Our system should record the acceleration transducer data of the movement and extract the pattern of the data. This pattern will be the criterion.
2. When the user want to use the device, them should do the pre-set movement to unlock the device. We need to record the acceleration transducer data of the new input movement and extract the pattern of the data again.
3. The system use the algorithm to judge that if the movement is the same as the pre-set movement by compare new input movement data pattern to the criterion one. For this part we can use Dynamic Time Warping (DTW), fast DTW and Hidden Markov model (HMM).
4. If the system judge that the movements are the same, we unlock the device; otherwise, we just keep it locked.

1.3. Objective

The objectives of our project is to find the most efficient algorithm to distinguish some specific movements of hand. There are many algorithms for machine learning, which can help us to deal with our problems. Because we want to implement this system on wearable device, we want the complexity of the algorithms to be as small as possible. So which is the best for our system? We use different algorithms to recognize the movement pattern and compare the performance of Dynamic Time Warping (DTW), fast DTW and Hidden Markov model (HMM). The time complexity is the most important, but we also to make a tradeoff between space complexity, size of training set and more influences.

1.4. Adopted approach

1. Retrieve data by using sensor (tri-axis accelerometer) on Moto 360.

2. Process the data by filter to get a pattern of the movement data.

3. Doing similarity analysis of standard pattern and new input pattern.

4. Build on training set.

5. Set balance point for similarity result.

1.5. Adopted algorithms

1.5.1. DTW:

In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation.

1.5.2. Fast DTW:

Fast DTW is an approximate Dynamic Time Warping (DTW) algorithm that provides optimal or near-optimal alignments with an O(N) time and memory complexity, in contrast to the O(N^2) requirement for the standard DTW algorithm. Fast DTW uses a multilevel approach that recursively projects a solution from a coarser resolution and refines the projected solution.

1.5.3. HMM:

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A HMM can be presented as the simplest dynamic Bayesian network. The mathematics behind the HMM was developed by L. E. Baum and coworkers.

In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

2. Methods / Results / Approach

What we do is using action recognition to unlock the wearable device. We get three-axis acceleration sensor data collected by Moto 360 device, then by analyzing different motion recognition algorithm time complexity and accuracy to determine which algorithm has better adaptability to our software.

2.1. Methods

We collect data through first Moto 360. Because users are easy to forget the gestures and movements of their own rhythm. When collecting data, we set up guidelines by vibration — once per second. Because identification gestures requires enough long length in order to ensure accuracy, and too long to collect for unlocking is unrealistic, we collected seven seconds sensor data.

What's more, we also consider the performance limitation of Moto 360 CPU, we need an algorithm which has a low time complexity and it does not need two much training sets. We choose Dynamic time warping algorithm and Hidden Markov model algorithm to implement our prototype.

We also think which sensor is better for us to do motion recognition, finally we find three-axis accelerometer is a good choice to implement it, reason are as following:

1. Every wearable device have three-axis accelerometer sensor.
2. Three-axis accelerometer sensor has more information than other sensor, it has three value for three axis(x, y and z).
3. Accelerometer data has more identify information than other sensor since motion's accelerometer data has big difference between each different motion.

When we doing our project, we have several challenge to finish this project, as followings:

1. We never learned Android and Java programing. When we collecting data, we have difficult in collect data from Android API.
2. We never learned design of UI, so we need to learn how to design it.
3. Although it have some paper which is discussing how to do motion recognition, but we just can find few reference about wearable device.

4. Because performance limitation of wearable device, we need to connect phone with Moto 360 to process sensor data.

2.2 DTW

2.2.1 Introduce to Algorithm

In our project, the best performance algorithm is the DTW.

DTW-dynamic time warping (DTW) is used to analysis and measure similarity between two temporal sequences which can be different in time or speed. The algorithm is based on dynamic programming (DP) of thinking to solve the problem of template matching pronunciation of varying lengths. For instance, we can use the DTW algorithm to calculate the similarities in walking patterns, even if they don’t have the same speed, or if there were accelerations and decelerations during the course of an observation.

And also it is the earlier, more classic kind of speech recognition algorithm appears for isolated word recognition. HMM training phase algorithm needs to provide a large amount of speech data, by repeatedly calculating the model parameters can be obtained, and DTW algorithm training almost no additional computation. Therefore, in isolated word speech recognition, DTW algorithm is still widely used. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data which can be turned into a linear sequence can be analyzed with DTW. A well-known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

In general, an optimal match between two given sequences (e.g. time series) with certain restrictions can be calculate by the DTW algorithm. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

2.2.2 Implementation of Algorithm

For example, here are two time sequence Q and C with the length n and m.

$$Q = q_1, q_2, \ldots, q_i \ldots q_n;$$
$$C = c_1, c_2, \ldots, c_j, \ldots c_m;$$

If n is not equal to m, using the linear amplifier to make the two sequence as the same length. But if n is equal to m, the only thing need to do is calculate the distance between two symbols.

For making these two sequence to have the same length, the first thing is to create an n*m matrix grid.

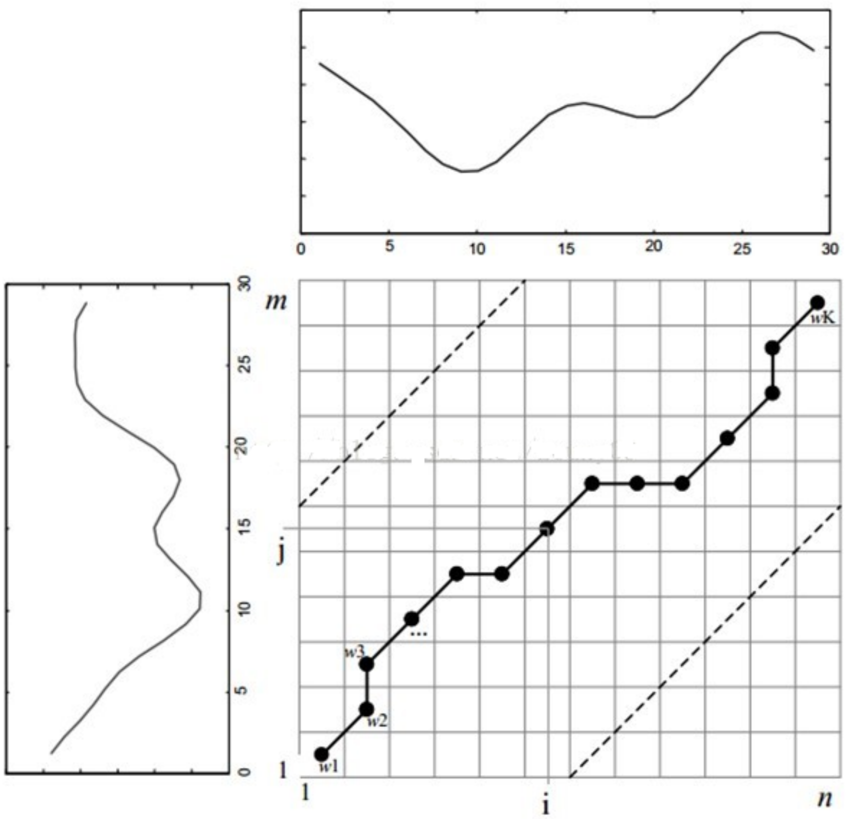

Figure1. N*m matrix of DTW

DP algorithm can be attributed to the grid through which to find a number of grid points of the path, the path through the lattice is calculated two sequences aligned point

$$W = w_1, w_2, \ldots, w_k, \ldots, w_n$$

The denominator of the K is mainly used for regular paths of different lengths to do compensation. What is our purpose? DTW or what thoughts are? The two time series is extended and shortened to give the shortest distance between two time series that is most similar to the one warping, the shortest distance is the distance metric last two time series. Here, we have to do is to choose a path, so that the total distance of the resulting minimum.

Here we define a cumulative distance cumulative distances. From (0, 0) point matching the two sequences Q and C, each to a point, before all of the points will accumulate from the calculations. After reaching the end of (n, m), the cumulative distance is what we said above, the final total of the distance, which is the sequence Q and C similarity.

Cumulative distance γ (i, j) can be expressed in the following manner, cumulative distance γ (i, j) for the current grid point distance d (i, j), Euclidean distance is the point of qi and cj (similarity) and can reach the minimum elements of the adjacent point and the cumulative distance:

2.3 Fast DTW

2.3.1 Introduce to Algorithm

Because the high time complexity of DTW, we decided to use Fast DTW to implement out app. The methods used make DTW faster fall into three categories:

a. Constraints – Limit the number of cells that are evaluated in the cost matrix.
b. Data Abstraction – Perform DTW on a reduced representation of the data.
c. Indexing – Use lower bounding functions to reduce the number of times DTW must be run during time series classification or clustering.

2.3.2 Implementation of Algorithm

Constraint is the most usual way to implement Fast DTW, two of the most commonly used constraints are the Sakoe-Chuba Band and the Itakura Parallelogram.

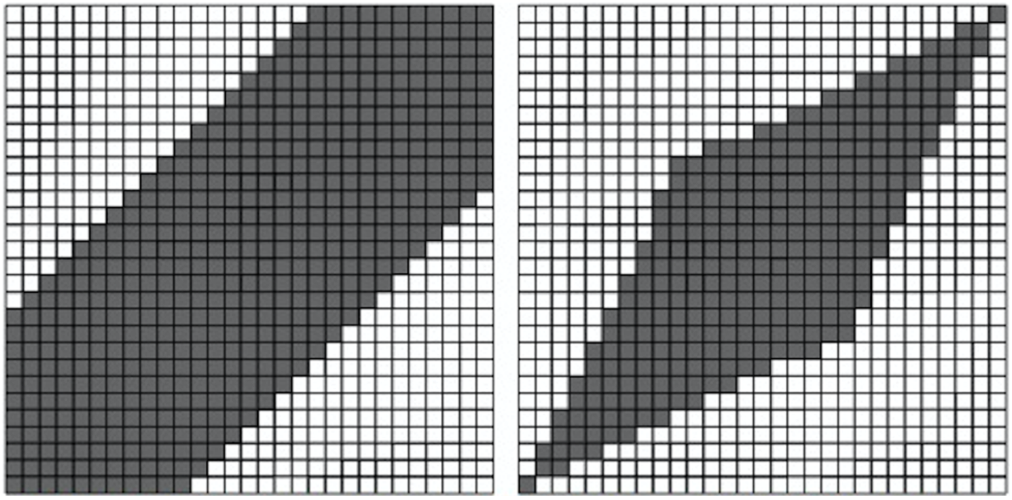

Figure 2. Sakoe-Chuba Band (left) and Itakura Parallelogram (right)

When constraints are used, DTW algorithm to find the optimal path by constraining the window. It is possible that it cannot found the optimal path because it do not calculate full matrix. By this way, we can expect the complexity of this is relatively straight line. Constraints work poorly if time series are of events that start and stop at radically different times because the warp path can stray very far from a linear warp and nearly the entire cost matrix must be evaluated to find the optimal warp path.

An another way to speed up DTW is using data abstraction. The figure as following is show it can cut matrix into several small matrixes, then calculate value in cell.

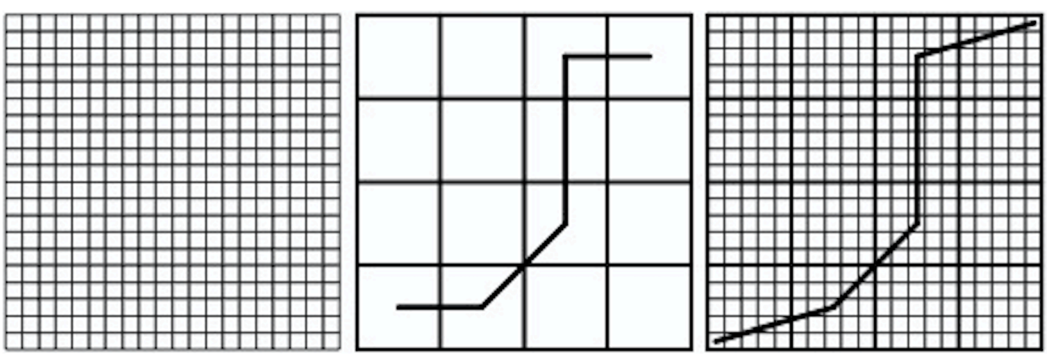

Figure 3. Speeding up by Data Abstraction

Indexing uses lower-bounding functions to prune out the number of times DTW needs to be run for certain tasks such as clustering a set of time series or finding the time series that is most similar to a given time series. Indexing significantly speeds up many DTW applications by reducing the number of times DTW is run, but does not speed up the actual DTW algorithm.

The accuracy of an approximate DTW algorithm can be measured by determining how much the approximate warp path distance. Error of FastDTW algorithm, function is as following:

$$\text{Error of a warp path} = \frac{approxDist - optimalDist}{optimalDist} \times 100$$

2.4. HMM

2.4.1 Introduce to Algorithm

HMM (Hidden Markov Model, HMM) is a statistical model, which is used to describe an unknown parameter contains implicit Markov process. The difficulty is to determine the process parameters can be observed from the hidden parameters. These parameters are then used to further analysis, such as pattern recognition.

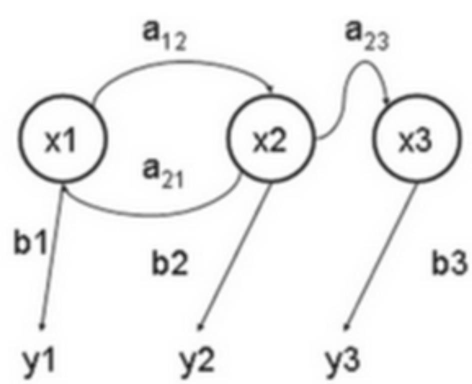

Figure 4. Pattern

Hidden Markov Model is thought to be modeled in the system is a Markov process with unobserved (hidden) state statistical Markov model.

Hidden Markov model is a Markov chain, its state cannot be observed directly, but through the observation vector sequence observed by each observation vector is some probability density distribution of the performance of the various states, each an observation is made having a state vector corresponding probability density distribution of sequence generation. Therefore, HMM is a double random process ---- a certain number of states of hidden Markov chains and show random function set. Since the 1980s, HMM is applied to speech recognition, a major success. In the 1990s, HMM also been introduced into computer text recognition and mobile communications core technology "multi-user detection." HMM in bioinformatics, fault diagnosis and other fields also began to be applied.

One kind of HMM may be presented as the simplest dynamic Bayesian network. HMM mathematics behind by LEBaum and his colleagues developed. It is early and optimal nonlinear filtering problems raised by the RuslanL.Stratonovich.

In a simple Markov models (such as Markov Chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In hidden Markov model, the state is not directly visible, but the output is dependent on the state, is visible. Output by each state may have a possible sign of the probability distribution. Therefore, to provide some information about the status of the sequence tag sequences generated by a HMM. Note that "hidden" means that the model passes through its state sequence, rather than the parameters of the model; even if these parameters are precisely known, we still regard the model is called a "hidden" Markov models. Hidden Markov model known for its pattern recognition over time, such as speech, handwriting, gesture recognition, marking parts of speech, music, partial discharges and bioinformatics applications.

Hidden Markov model can be considered a hybrid model outlined in the hidden variable (or variable), which controls the mixed ingredients are selected for each observation, by a Markov process rather than independently correlated. Recently, hidden Markov model has been extended to every two triplet Markov models and Markov models allow for consideration and non-stationary data modeling more complex data structures.

2.4.2 Implementation of Algorithm

Hidden Markov Models (HMM) can be used to describe the five elements, including two state sets and three probability matrix:

1. The implied state S

Satisfy the Markov property between these states, Markov model actually implied state. These states are not usually obtained by direct observation. (E.g. S1, S2, S3, etc.)

2. Observable state O

Implicit in the model and associated state, can be obtained by direct observation. (Such as the number O1, O2, O3, etc., the state does not have to be observed and the number of hidden state of the same.)

3. Initial state probability matrix π

Implicit in the initial state represents time t = 1 the probability matrix (e.g., when t = 1, P (S1) = p1, P (S2) = P2, P (S3) = p3, the initial state probability matrix π = [p1 p2 p3].

4.The implicit state transition probability matrix A.

It describes the transition probability of each HMM model states.

Where T represents the time, the state is under the conditions of Si, in time t + 1 is the probability Sj of state.

5. Observing the state transition probability matrix B (the English called Confusion Matrix, literally translated as confusion matrix is not easy to understand from literally).

Let N be the number of state representatives implied, M representative of the number of states can be observed, then:

$$B_{ij} = P(O_i \mid S_j), 1 \le i \le M, 1 \le j \le N$$

T represents the time, the state is under Sj implied conditions, the probability of Oi's observation state.

Summary: In general, you can use $\lambda = (A, B, \pi)$ triples to concisely represent a hidden Markov model. Hidden Markov model is actually an extension of the standard Markov model, adding a set of states can be observed and the relationship of these states and implied probability between states.

2.5 Motion

For our data collecting, we record 7 seconds of movement, with 100 data per second. Because we collecting data of the three-axis accelerator, so we got data for x-axis, y-axis, and z-axis at the same time.

There are some thresholds in our algorithms, which decide the movements are the same or not. If we set a large range, then some time the algorithm will think a wrong movement is as same as the standard one; but if we set a small range, the algorithm will decide a right movement is not as same as the standard one because of some environment influence. In order to decide a best value of these thresholds, we need to set up a training set for them.

There are three kinds of setting movement of our training set:

1. Repeating one times of up and down per second for 7 seconds.
2. Go up, down, and up; left, right, and left; up, down, and up; left. Each action per second.
3. Draw an arc per second, from right to left, like drawing a mouth of a smiling face. Repeating for 7 seconds.

Every team member use a different kind movement and did 25 times of hand movements. 20 of them are repeating the setting one, 5 of them are some random actions. Then we use the data of these movements to do the comparison. We want to see when we do the same movement, the result would be in what kind of range; and if we do different movement, the result should be in different range. Using these result, we come up with the thresholds of different algorithms.

2.6 Use of Standards

Some standards use in design:

a. Standardized network technologies:

Bluetooth, IEEE 802

b. Open source standards:

   Android Wearable Device

2.7 Experiment / Product Results

```
C:\Users\xiaopei zhang\Documents\Visual Studio 2013\Projects\DTW\Debug>DTW.exe
find the file
data processing...
the total time of quick find are 9 ms
x-axis5.37
y-axis1.096
z-axis3.402
the similiarity of these two sequence3.28933
C:\Users\xiaopei zhang\Documents\Visual Studio 2013\Projects\DTW\Debug>DTW.exe
find the file
data processing...
the total time of quick find are 9 ms
x-axis0.594
y-axis0.677
z-axis2.242
the similiarity of these two sequence1.171
C:\Users\xiaopei zhang\Documents\Visual Studio 2013\Projects\DTW\Debug>DTW.exe
find the file
data processing...
the total time of quick find are 8 ms
x-axis5.45
y-axis4.99
z-axis4.683
the similiarity of these two sequence5.041
C:\Users\xiaopei zhang\Documents\Visual Studio 2013\Projects\DTW\Debug>DTW.exe
find the file
data processing...
the total time of quick find are 9 ms
x-axis5.84
y-axis0.741
z-axis0.96
the similiarity of these two sequence2.51367
C:\Users\xiaopei zhang\Documents\Visual Studio 2013\Projects\DTW\Debug>DTW.exe
find the file
data processing...
the total time of quick find are 12 ms
x-axis7.353
y-axis0.788
z-axis1.28
the similiarity of these two sequence3.14033
```

Figure 5.Value after similarity calculate

We have run 20 data set from every volunteer, and we create 10 compare datasets:

DTW		Fast DTW		HMM	
Average Processing Time of 7 seconds data	Accuracy (%)	Average Processing Time of 7 seconds data	Accuracy (%)	Average Processing Time of 7 seconds data	Accuracy (%)
9 ms	96	3 ms	74.7	578ms(training sets)+9ms(similarity analysis)	97.5

Table 1. Processing Time and Accuracy of DTW, Fast DTW & HMM algorithm

Because we just use 20 datasets to do hmm algorithm and accuracy is quite high when we using DTW algorithm. HMM need to compare each datasets to set training sets, so it need to compare n+(n-1)+…+1 datasets to finish set training dataset.

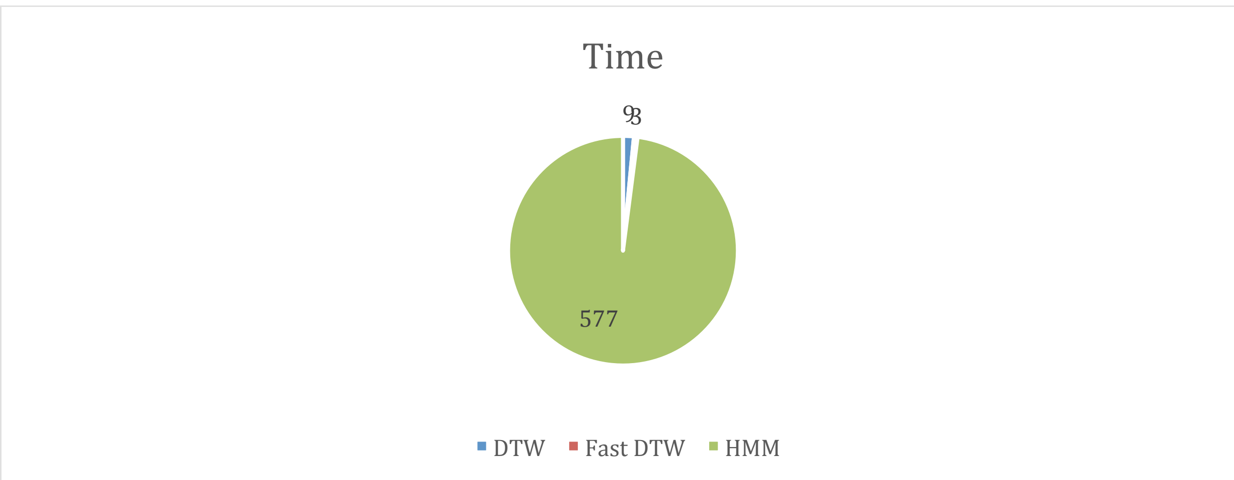

Figure 6. Time of DTW, Fast DTW & HMM algorithm

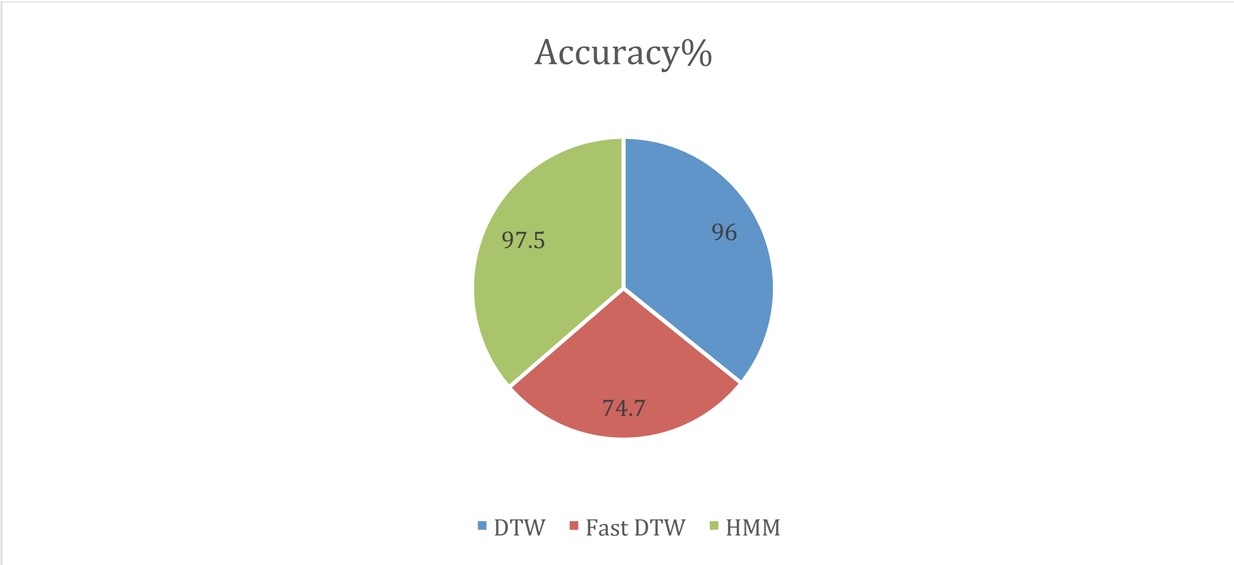

Figure 7. Accuracy of DTW, Fast DTW & HMM algorithm

As we can see above, Fast DTW has very low accuracy because it do not compare every point of data set.

3 Cost and Sustainability Analysis

3.1Economics impact

According to Forbes, 71% of 16-to-24 year olds want wearable tech. In long term, the wearable device market is massive, no doubt. But not many people use them now. The reason is that the wearable devices' functions are too limited now. Most of the time, we can only get some basic functions from them, such like the heart rate of user, get a notification from the cell phone, or just use them as a pedometer. People do not want spend too much money on them. The wearable devices are in sore need of comprehensive features, and security is an important part of it.

If we develop a way to lock and unlock the wearable, or even go further, we can ask the wearable device to unlock our cell phone, we can greatly enhance the practicability of wearable device. After using our system, they can check short massage or e-mail on their MOTO 360 watch without worrying about other people do this before you agree. When the wearable devices become more fully functional, there will be more people willing to buy and use them, then the market will expand and people can enjoy a more convenient life with these devices.

3.2 Environmental impact

Our system do not have any hardware, so there is no cost on material.

We compare the performance of Dynamic Time Warping (DTW), fast DTW and Hidden Markov model (HMM). When we use the most efficient algorithm, then we can use less time on processing. By doing this, we can also use less energy and cut down the cost of this part.

3.3 Social impact

People want to spend less space on carrying different kinds of devices, or another way to deal with it is just wear them on. This is an important property of wearable devices. Our system can enhance the function of wearable device. After improved the performance of them, people can spend less time on open the bag, find the smart phone, and check the e-mail. We can just use few seconds moving our hand, and the message will show on MOTO 360, just like have a look at the watch. Everything will become more easy and convenient. This system help people to simplify the way to keep in touch with our friends.

4. Conclusions / Summary

We would like to find an algorithm to distinguish some specific movements of hand. After compare Dynamic Time Warping (DTW), fast DTW and Hidden Markov model (HMM), we find that DTW is the best for time complexity and accuracy. Fast DTW will use less part of data as the pattern, so it is fast but poor in accuracy. HMM need a large data set for training, which is more complicate to implement with little improvement of accuracy. So we choose DTW as our algorithm to recognize if the user input the right hand movement and if we should unlock for him/her.

In the future, we may try more kinds of algorithm, such like Support Vector Machine (SVM), and to see if we can get better results for our system.

5. Acknowledgments

We would like to thank Prof.Zhang, Shugang Li, and Yihan Qian.

6. References

[1] Dynamic time warping for speech recognition with training part to reduce the computation Sun Xihao, Yoshikazi Miyanaga

[2] Dynamic time warping for speech recognition with training part to reduce the computation Sun Xihao, Y. Miyanaga

[3] An improved dynamic time warping algorithm employing nonlinear median filtering Yoshikazu Miyanaga

[4] Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques L Muda, M Begam, I Elamvazuthi