Waseda University Master Thesis

Authentication System using Hand Gesture and Depth Camera

44151576-4: Jinghao ZHAO

Master (Engineering)

Supervisor: Professor Jiro TANAKA

Interactive Programming
Information Architecture
The Graduate School of Information, Production and Systems

July 2017

ABSTRACT

With the development of computer science, humans are concerned more about their privacy, traditional text password becomes weaker to defend from various attacks, meanwhile, somatosensory become popular, which makes gesture authentication become possible. Traditional methods are mainly based on computer vision technology or wearable sensor embedded in wristbands or smartphone, which will cost a lot on equipments, or lost accuracy because of devices. Biometric technology bring new chance to improve the authentication method. Therefore, this research is try to use humans dynamic hand gesture to make an authentication system, which should have low limitation and be natural. In this thesis, we described a depth camera based dynamic hand gesture authentication method, and also give some evaluation about this method.

Keywords: Gesture Authentication, Three-Dimensional Hand Gesture, Depth Camera

ACKNOWLEDGMENTS

The thesis and related experiments are written by the guidance of Prof. Jiro TANAKA, professor is rigorous on study, also works carefully. He gave us a lot of valuable advice so that we can finished the research in a limited time. Here, I would like to express my heartfelt thanks to Prof. TANAKA for his support and help in the experiment.

My lab members gave me a lot of experimental ideas and technical support, so that we can successfully complete the experiment and get good experimental results, I would like to express my thanks for their warm help.

I would like to thank my parents and loved ones for their encouragement and support in both spiritual and material conditions, It is the support that let me complete my master studies. The only thing I can do is keeping moving forward and making progress, getting better results to return them!

Contents

List of Figures

List of Tables

Chapter 1

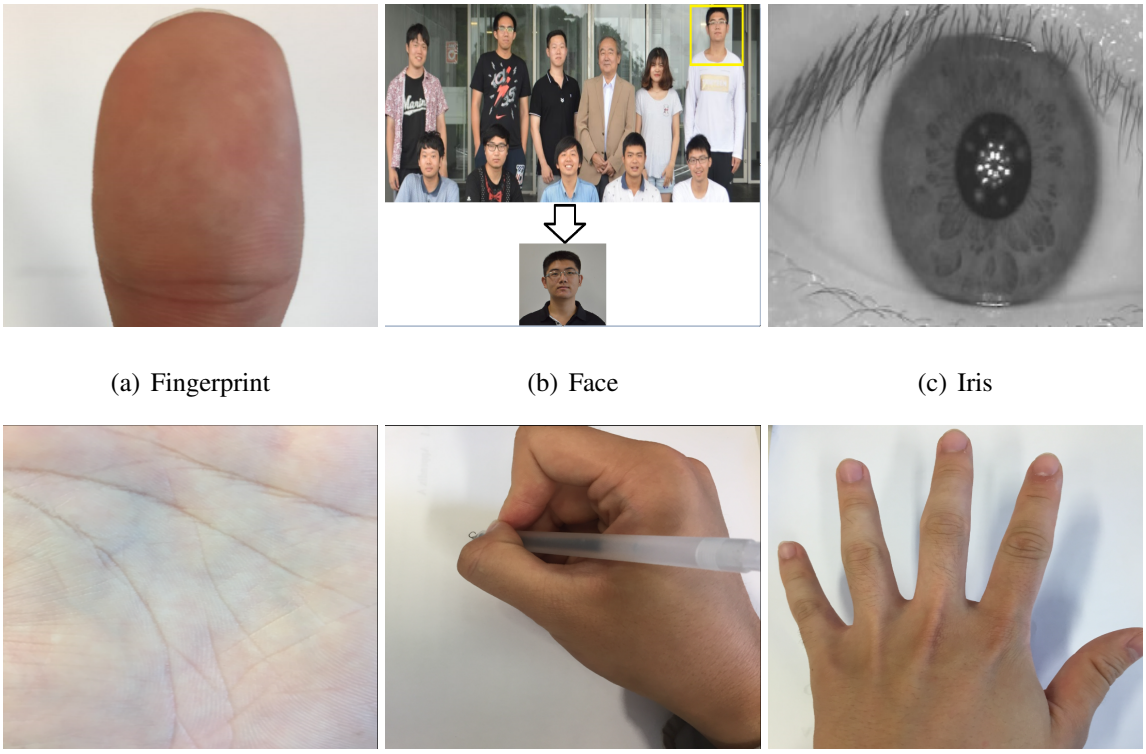
Introduction

1.1 Introduction

In recent years, as information technology are becoming rapidly increasing, computer and mobile phone become unique part of human life, which have a deep relation with user's privacy and information. How to protect such kind of privacy attracted the attention of several aspects. Text password is the most frequently way that we used in our delay life, but recent research shows that such kind of password become weaker and need to be longer and more complex to keep its performance[1], which will make it very inconvenient and unnatural. Besides, even we used such kinds of text password, it also can be easily stolen or copied by smudge attack or shoulder surfing attack[2, 3]. Therefore, how to use new methods to replace the traditional text password become more important in security and human computer interaction fields.

In order to make up for the shortage of traditional text password, biometric information is used as a new kind of authentication methods. As a new kind of access control information, biometric information consist s of physical characteristics and behavioral characteristics. Physical characteristics are referred to the static features of human body, such as iris, fingerprints or DNA, behavioral characteristics means dynamic information from human's behave, including gait, gesture, typing

and so on. According to the source of biometric information, it also can be devided into congenital features and acquired features.



(a) Fingerprint (b) Face (c) Iris

(d) Palm (e) Handwriting (f) Shape

Figure 1.1 Several different biometric recognition methods

Current biometric authentication technologies are mainly based on computer vision technology with RGB camera or wearable sensor such as accelerometer embedded in wristbands or smart phone, those methods have already achieved good results, but both of them still have demerits. The computer vision technology can get high accuracy, but needs to deal with a large amount of image data, which cost time on computing. Meanwhile, it has high demands on equipments, users are always need to perform in special environment. The wearable sensor method is portable and usually does not have environment limitation, but it has some accuracy lost because of sensor itself, and some sensors have difficulty in power charging. Hence, how to combine the good points of

each method to make a more natural authentication system become more important.

In this thesis, we mainly did research on following aspects:

1. Described a Leap Motion based hand gesture authentication method. We mainly used dynamic behavioral gesture features to built our system.

2. Came up with a series of processes to deal with Leap Motion data and used those data to identify a user.

3. Gave the evaluation of this authentication methods based on False Acceptance Rate(FAR) and False Rejection Rate(FRR).

1.2 Organization of the thesis

The structure of this paper is designed as follows: Chapter 2 introduces some basic background and related work on such field. Chapter 3 given the goal and approach of our research. In Chapter 4, we describe our system prototype and methods of the hand gesture authentication system in detail. Chapter 5 will give the implementation of our system with listed methods. Then, in order to evaluate the proposed method and testify its effectiveness, Chapter 6 evaluates our system by some accuracy experiment with and without template updating mechanism. Finally , Chapter 7 gives the conclusion and discussion of our work.

Chapter 2

Background

We will introduce the related work in several aspects: first, we discussed some biometric techniques and application. Then introduce some authentication using hand gesture. Finally, we will give some research for evaluating an authentication system.

2.1 Depth Camera

The 3D depth camera is a kind of new stereoscopic vision sensor with three-dimensional depth sensing module, which can get both RGB and depth data with high precision and resolution in a low time-delay, building 3D image timely. It can be used in real0-time target target recognition, motion capture or scene perception for 3D images. Depth information is stable and reliable, and is not affected by ambient light. Now it is often used as embedded modules in the smart TV, smart phones, smart home appliances, 3D printing, wearable equipment, machine vision to achieve deep perception and human-computer interaction.

Leap Motion is used as the depth sensor in our system. Leap Motion is a kind of depth camera that can track hands and fingers. It is a small USB peripheral device which is designed to be placed on a physical desktop, facing upward. It can also be mounted onto a virtual reality headset. It

has two embedded cameras and three infrared LEDs, the device can sense an inverted cone area, which has about one meter range. The IR cameras return about 200 frames per second, each frame contains various hand and finger information. Figure 2.1 and 2.2 shows the structure of Leap Motion.

Figure 2.1 Leap Motion Real view

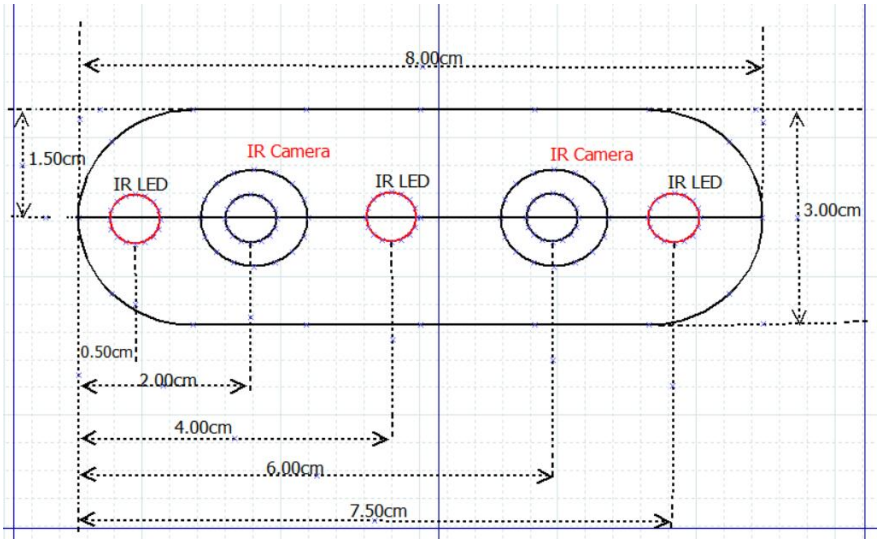


Figure 2.2 Leap Motion Schematic View

The Leap Motion controller has a right-handed Cartesian coordinate system and works with optical sensors and infrared light. Figure 2.3 is the coordinate built by Leap Motion.

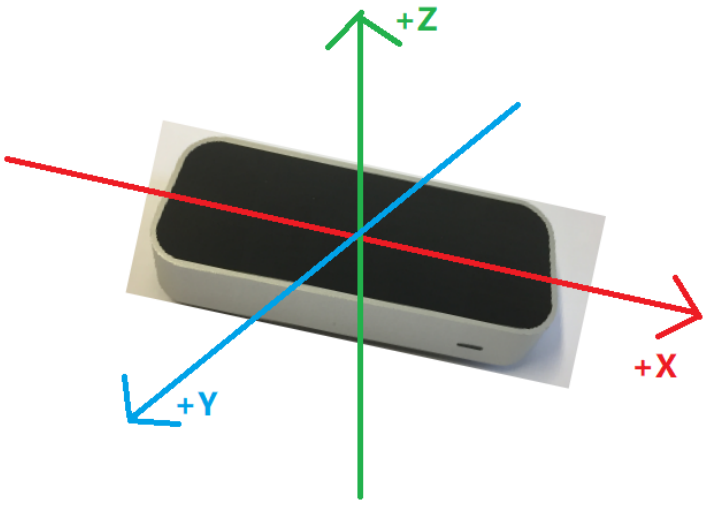


Figure 2.3 Leap Motion Coordinate

Weichert et al.[4] discussed the performance of Leap Motion on accuracy and robustness. With test of tracking a subject and get the deviation of real position and Leap Motion data, the authors got that the average deviation is less than $0.2mm$ in static case, and a $0.7mm$ deviation in dynamic case. Although it is difficult to achieve the theoretical accuracy of $0.01mm$ under real conditions, compared with Microsoft Kinect and other depth camera in gesture-based user interfaces, it is still have a relatively high precision.

As a new kind of depth camera, Leap Motion has following merits :

1. It works in an intimate proximity with high precision, tracking position and motion for a millimeter level gap.

2. It can works on various operating system, also developed by different programming languages.

3. It has a smaller size than other depth cameras.

2.2 Biometric Techniques

There are several researches have got progress on Biometric authentication, famous technologies such as iris and fingerprint recognition are widely used. Daugman[5] came up with iris authentication method for recognizing persons. Maltoni et al.[6] introduced a kind of fingerprint recognition methods, which is widely used in some company. Chen et al.[7] generate a rhythm-based authentication system by using the rhythm when user click or touch the screen of tablet .As for behavioral biometric characteristics, Gafurov et al.[8] used accelerometer as the sensor and came up with a gait authentication method. Other biometric such as voice, face recognition, vein scanning and so on are also used as authentication methods, but most of them are complex or need high equipment cost, static physical characteristics even have high possibility to be copied or stolen by attackers. However, our method have low cost on devices, and we have better security performance because we used dynamic behavioral features.

2.3 Hand gesture authentication

As an unique biometric information for individuals, hand gesture is proved based on several research that it can have good performance in authentication system. Liu et al. presented an accelerometer-based personalized gesture recognition methods[9], in their paper, acceleration data of hand gesture is sensed by mobile phone, and they used Dynamic Time Warping(DTW) to analysis and match the input gesture. Kratz et al.[10] presented an air-auth hand gesture authentication system using Intel Creative Senz3D, they also made some improvements and assessment methods. Chahar et al. used Leap Motion and Neural Network built a dynamic hand gesture authentication system[11]. Those researches shows that dynamic hand gesture could obtain high accuracy in authentication.

2.4 General Tools

2.4.1 Computer Vision based Methods

The computer vision method is to use the camera and computer to monitor, track and identify the target. The image is then further processed to make it more suitable for transmission to the instrument to detect images or directly observed images, such as X-ray security machine image processing. This methods is a more mature way in identity, but there will be some shortcomings

1. It is difficult to extract data from video or image stream, sometimes it need to deal with large image data;

2. The change of external environment may cause some decrease on accuracy.

3. It is difficult to evaluate the algorithm.

2.4.2 Wearable Sensor based Method

With the development of sensor technology, Many sophisticated sensors are used for identification such as RFID sensor, accelerometer or gyro sensor. Such kinds of sensor are embedded in wristband or belt or other carry items, and sense data timely. It is more convenient and freedom, but some sensor are not as accurate as computer vision device, meanwhile, some sensors have problem with power charging, which limits the work hours.

2.5 Evaluation methods

Jain et al. have proposed seven basic rules for an authentication system[12], which are universality, uniqueness, permanence, measurability for any biometric system and performance, acceptability, circumvention in a practical biometric system. Aumi et al.[13] discussed Trajectory security performance of their system by the test of shoulder-surfing attacks and smudge attacks.

Yang et al.[14] discussed how to evaluate the permanence and memorability of their research. We recorded the accuracy of such classification system in following table 2.1.

Table 2.1 Comparison of 333related authentication system

Reference	**Sensors**	**Features**	**Classifiers**	**Accuracy**
[13]	Intel Senze3D	trajectory	DTW	94.3%
[15]	Leap Motion	shape & path & speed	threshold	not mentioned
[9]	3D accelerometer	acceleration of fingertip	DTW	98.4%
[7]	tablet	shape & pressure & distance	SVM	95%
[16, 14]	tablet	trajectory	threshold	91.67%
[17]	smartphone accelerometer	acceleration	DTW	93% - 97%

2.6 Related Work

Chahar et al[11] used Leap Motion and hand gesture built a Leap password system. In their work, they proposed a aLPhabet framework which used hand and finger static shape data such as length and width information, combined with time information for each user to perform a whole gesture and verify user's identity. In their work, they used Levenshtein Algorithm to estimate the similarity for gestures,and gave weight to each kind of feature to ranking the importance of features, finally, they used Naive Bayes, Neural Network and Random Decision Forest classifier and get an average score for each possibility and get the classification result. In their work, they kept a 1% FAR, and got an accuracy about 81%.

Compared with Chahar's work, our system got a relative high accuracy with some lost on FAR, meanwhile, we used dynamic hand gesture in our system, which is more unique and more difficult to be copied by tools, and we did some attack and safety experiments to prove the stable of our system.

Aumi et al[13] used dynamic hand gesture and Intel Senze 3D in their work, and they used Dynamic Time Warping(DTW) as their classification method. They also did some attackn experiments such as shoulder-surfing threaten, and got a high accuracy if they set a very low threshold for DTW. Besides, they designed a template updating mechanism by counting successful access times.

Compared with Aumi's work, we used Leap Motion, which is lighter and easier to use. Meanwhile, we proposed a double threshold template updating mechanism based on period and access times, which can reduce the threaten of false acceptance.

Chapter 3

Research Goal and Approach

In our research, we are aimed to find a robust authentication methods by using human dynamic hand gesture. Since traditional text password are not friendly to users, while some biometric authentication methods such as iris and fingerprint methods need high requirements on equipments and can not be pervasive. Therefore, we try to find a natural and acceptable authentication method for users, also, it should have relative high performance.

To achieve our goals, we did research on different aspects. We used new generation depth sensor - Leap Motion, which has lighter weight but keep a high accuracy. In order to get a good result, we discussed a series of preprocessing methods on smoothing and normalization. Then, we came up a feature extraction method based on clustering and filtering. Finally, we used Hidden Markov Model(HMM) to solve the classification of identities.

Instead of using static hand shape or other static biometric information, we used dynamic hand gesture as our core elements for authentication. Because dynamic gesture contains more information than static posture, it is harder to be copied or imitated by others, meanwhile, it has lower equipment requirements than traditional fingerprint or iris authentication methods, which will make it can be used in more situation.

Chapter 4

System Design

The authentication based on depth camera is a combination of signal processing, pattern recognition, machine learning and other field. It needs to solve the problem of obtaining data and analyzing data. It usually includes data preprocessing, classification and recognition.

4.1 Data Preprocessing

4.1.1 Data Smooth

Because their are noises remains in the raw data, we used Kalman filter to eliminate those noises. Kalman filter is a recursive filter that works well on one-dimensional linear systems with Gaussian error statistics[18]. The basic idea of Kalman filter is that current state s at time t is related to previous state at time $t-1$. The purpose is to find the optimal value for all states in time series.

In processing control, suppose we have a system which can be described in a following linear stochastic difference equation,

$$s_t = As_{t-1} + BU(t) + C \tag{4.1}$$

where A and B is system parameter of this system. s_t is the feature data of time t, we called it state

in this algorithm. $U(t)$ is the external control, and C is the measurement noises from devices which can not be measured. What we can measure is the output of device.

$$y_t = Hs_t + N_t \tag{4.2}$$

where y is the output value, H is parameter of system, and N is the noise in reading data from device. Obviously, the output value contains two part of noises, one is measurement error C, another is reading error N. We use Q and R to represent the covariance of C and N. To erase these two error, Kalman filter used two steps, prediction and measurement[19]. Prediction is predicting current value based on previous state, and measurement is modify the observable result of devices by using the covariance.

First we should use (4.1) to predict current state. In prediction step, the noises C are treated as 0 because it is try to predict the real value without noises.

$$s_{t|t-1} = As_{t-1} + BU(k) \tag{4.3}$$

where $s_{t|t-1}$ is the prediction result of time t based on state of time $t-1$, and s_{t-1} is the optimal result of time $t-1$. If there are no external control, B can be 0.

Next we should update the covariance of current state,

$$P_t = AP_{t-1}A^T + Q \tag{4.4}$$

Here, P_t is the covariances of current state, P_{t-1} is the prediction covariances of time $t-1$, Q is measurement covariance, A^T is the transpose matrix of A. Now we have the prediction of current state, then we should combine the observable value to get the optimal value.

$$s_t = s_{t|t-1} + Kg_t(y_t - Hs_{t|t-1}) \tag{4.5}$$

where Kg_t is the Kalman gain of time t, it is calculated by the covariance of measurement and prediction.

$$Kg_t = P_tH^T(HP_tH^T + N)^{-1} \tag{4.6}$$

then we got the optimal value of current time, finally, in order to make the filter work for next state, we should update the covariance of current time.

$$P_t = Kg_t HP_t \tag{4.7}$$

Therefore, the whole smooth algorithm is works as following pseudocode shows.

Algorithm 1 Kalman Filter Algorithm

Input: Initial state x_0, state sequence X;

Output: Smoothed sequence of X;

1: **function** KALMAN_FILTER($s_{t-1}, P_{t-1}, s_t, y_t$)
2: Use (4.3) to estimate current state value based on previous smoothed value.
3: Use (4.4) to calculate the covariance of prediction value.
4: Use (4.6) to calculate the Kalman gain.
5: Use (4.5) to get the current optimal value by estimated value $s_{t|t-1}$ and observed value y_t.
6: Use (4.7) to update the covariance to be used in next iteration.
7: **return** Current smoothed value f_t, current covariance P_t
8: **end function**

After that, we finished a calculation for smoothing data of time t, the whole smooth process will make an iteration until calculated the last value of this sequence.

Other researchers used different methods in their research, Nowlan et al.[20] used IIR2 filter with a length of 150 Gaussian window, Gafurov[8] used Histogram approximation to smooth the data, such kinds of methods also can achieve good result.

4.1.2 Data Normalization

In statistics and applications of statistics, normalization of ratings means adjusting values measured on different scales to a notionally common scale. we used normalization to make authentication data easy to match the template data. In our research, normalization is used to make it

easier to match the data from different gesture input. Since the input gestures may have difference in coordinate, normalization is needed to make the coordinate value normalized in the range of (-1,1).

There are various normalization methods in statistics, we used the standard score to calculate the normalized data of features information.

$$nx_i = \frac{x_i - \bar{x}}{\sigma x} \tag{4.8}$$

where $\bar{x}$ means the mean of smoothed x sequence, x_i means the value of time i and σx means the standard deviation of smoothed x sequence.

4.2 Feature Extraction

After we preprocessed with the raw information, we need to choose appropriate features to be used in classification. This step need to be discussed because if we used too few features, the uniqueness of system will dropped, but if we used too much features, the permanence of system may be lost. Besides, It is concluded that not all of the features have obvious affect on the authentication result. Therefore, we can only extract the useful features and ignore other features. Then we can ensure the reliability of its data while avoid large data operations.

Feature extraction also have deep influence on the result of classification. There are two main error existed in the using of classifier, one is empirical error, which means the error for classifier to learn and build model from training set, it may not classify all training set data as their true value; another is generalization error, which means the classifier has some error to classify new test data. And two main problems will be caused by two error: one is overfitting, which means the classifier consider too many features, and be "too strict" for new test data; another is underfitting, which means the features considered are not sufficient to complete the classification, the classifier are "too tolerant" for new test data. Figure 4.1 gave the intuitive comparison of overfitting and underfitting.

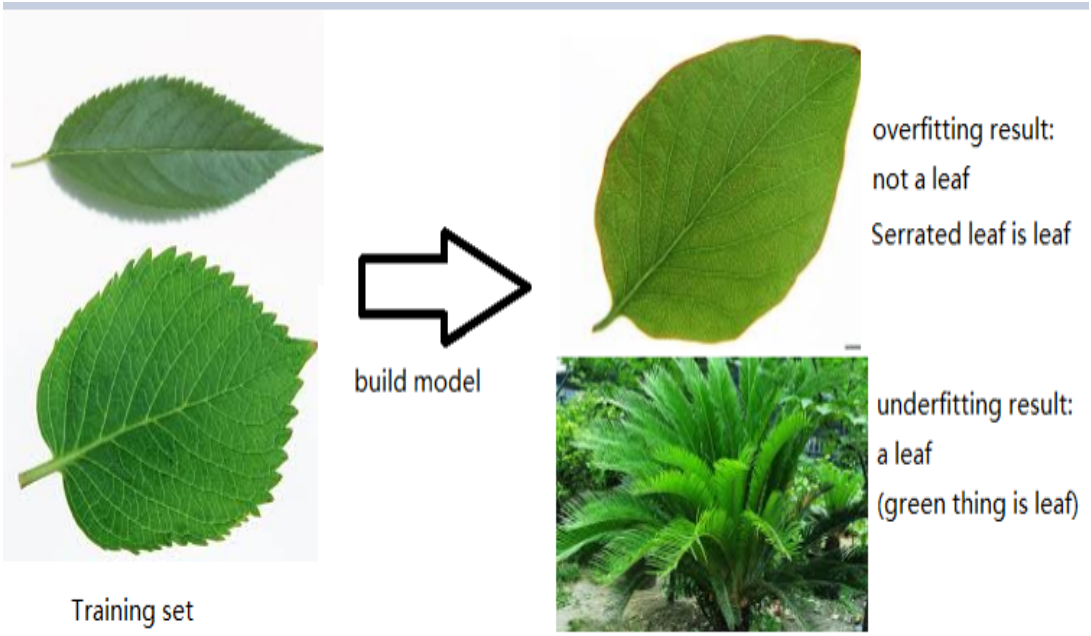


Figure 4.1 Example of underfitting and overfitting

In order to control variables in our system, we need to same features for different users. That will requires that the selected features should work well with most of users. And in order to reduce the risk of underfitting and overfitting problem, we used a kind of cluster method to select features we need.

4.3 Classification

There are several methods can be used to deal with time series gesture data, Dynamic Time Warping and Naive Bayesian are often used in related research. Our system used Hidden Markov Model to classify the user's identity. Hidden Markov Model (HMM) is a kind of simplest dynamic Bayesian network, it is mainly used in analyzing time series data. Hidden Markov Model have two kinds of variable, one is hidden state, another is visible output, which is depended on the hidden state and transition probabilities. In a Hidden Markov Model, the system is always transfered in different states$\{s_1, s_2, ..., s_N\}$[21].

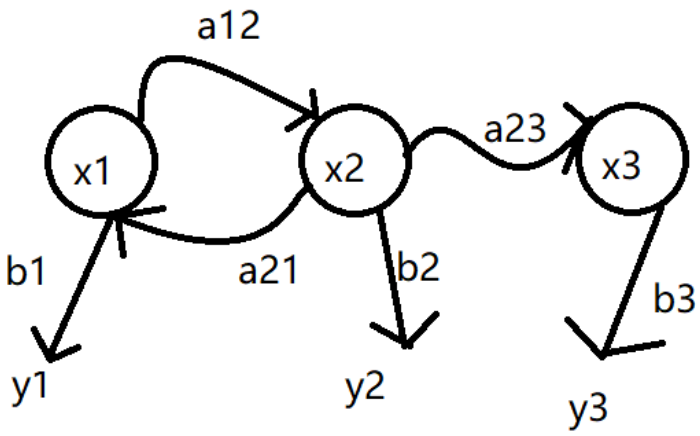


Figure 4.2 A Hidden Markov Model

Figure 4.2 is a simple example of Hidden Markov Model, the meaning of parameters are

1. X means hidden states of this model, which is only decided by previous one state.

2. y means possible observations, also can be the output of the model.

3. a is the state transition probability, which means the probability that the model will changed from one to another state.

$$a_{ij} = P(x_{t+1} = s_j | x_t = s_i), x_i, x_j \in X \tag{4.9}$$

which means, the probability that current state will be converted from s_i to s_j at any time t.

4. b means output probability, it is the possibility of each result that the system could get based on current state.

$$b_{ij} = P(y_t = o_j | x_t = s_i) \tag{4.10}$$

which represents that probability that result o_j will be observed under current state is s_i.

Also, there will have a initial state probability that decided which state will be the initial state of the model.

The basic conception of HMM is that observation is only depended on current hidden state, and it has no relation with other observation result or hidden state. Meanwhile, hidden state y_t at time t is only rely on hidden state y_{t-1} at previous time t-1. Based on this Markov chain, the joint probability distribution of all variable is

$$\begin{gathered} P(x_1, y_1, ..., x_n, y_n) = \\ P(x_1)P(y_1|x_1)\prod_{i=2}^{n} P(y_i|y_{i-1})P(y_i|x_i) \end{gathered} \tag{4.11}$$

There are three kinds of problem can be solved by HMM algorithm.

1. First is evaluation problem, which is calculate the possibility for generating a sequence of state based on transition probability a_{ij} and output probability bij;

2. Second is decoding problem, when we have a HMM model and a observation output, we can get the optimal hidden state which can generate this output;

3. Third is learning problem, it is used when we only know a general structure of a HMM, such as the number of states, but the possibility is not clear, then we can build a whole HMM model by using known dataset.

We used HMM to solve the third problem, the algorithm is as Algorithm 2 shows.

Algorithm 2 HMM Forward Algorithm

Input: Training set V,Convergence criteria θ , $t \leftarrow 0$

Output: a_{ij},b_{ij} HMM model that can recognize new data.

1: **function** FORWARD-BACKWARD ALGORITHM(t)
2: **while** $\max[a_{ij}(t) - a_{ij}(t-1), b_{jk}(t) - b_{jk}(t-1)] < \theta$ **do**
3: t ← t + 1
4: Calculate $a_{ij}(t)$ based on (4.9) and (4.11);
5: Calculate $b_{jk}(t)$ based on (4.10) and (4.11);
6: **end while**
7: **return** a_{ij},b_{ij}
8: **end function**

And we will build the HMM model by transition possibility and output possibility between different state.

4.4 Template Updating

Since hand gesture information can be changed with time passing, the registered gesture may not be durable. Therefore, template updating mechanism should be designed, previous work have already designed several kinds of updating mechanism[9] and proved to be effective.

The basic idea of previous work is when a user is successfully accepted by the system, the input gesture data will be stored in the template record, and used in next authentication. But this methods will remains some risk that it may updated at a false acceptance situation, therefore, we came up with our method to make improvement on it.

Our method works in following order:

1. Every gesture record will be created with a timestamp and will be used to build template or

authentication.

2. When gesture is used to build template, the latest gesture timestamp will be the timestamp of template. When gesture is used for authentication, if it is passed, it will be stored in a temporary gesture record.

3. We set both time threshold and record size threshold and they work independently. if the length of temporary gesture record reaches the size threshold, or the time interval from latest template to present time exceeds the time threshold, temporary records and templates will be merged to form a new template of this user.

Chapter 5

System Implementation

5.1 Prototype design

Leap Motion is used as our hardware platform. We built our system with following prototype 5.1.

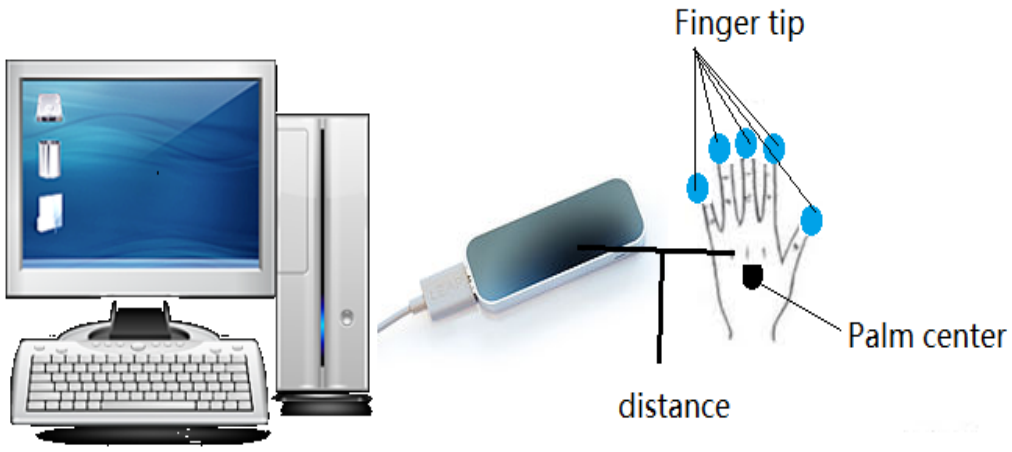


Figure 5.1 System Prototype

First, Leap Motion should works by connecting to computer, then, hand model can be automat-

ically build by it. Hand information will be send to API in real time in the format we need. The all tools are as follows

1. Leap Motion Controller
2. USB Cables
3. Computer with programming environment

And the minimum system requirements for the computer is:

1. Windows 7+
2. AMD Phenom^{TM} II or Intel Core^{TM} i3/i5/i7 processor
3. 2GB RAM
4. USB 2.0 port

From above we can see the equipment cost of Leap Motion is very low, we used a DELL Desktop with Inter(R) Core(TM) i7-6700 CPU and 8.00 GB RAM is also used in our system, the operating system is Windows 10 x64.

The system will works in following process:

1. First, users are asked to perform their gesture over Leap Motion camera to register their account and "gestural password", all of registered data will be stored as user's own template.
2. Then, when a user want to access his account, just choose his account and perform the registered gesture, the system will automatically return whether current user can pass or not. Passed verification gesture will be stored as a temporary template waiting to be used to update current template.
3. After a period of time, the accuracy of current template will dropped, therefore, the system will merge the temporary template with current template and make it become a new template.

4. Some times users may frequently access their account, the system will also update the template when successful access times achieved the threshold.

And the flow char 5.2 also shows the sequence of system work.

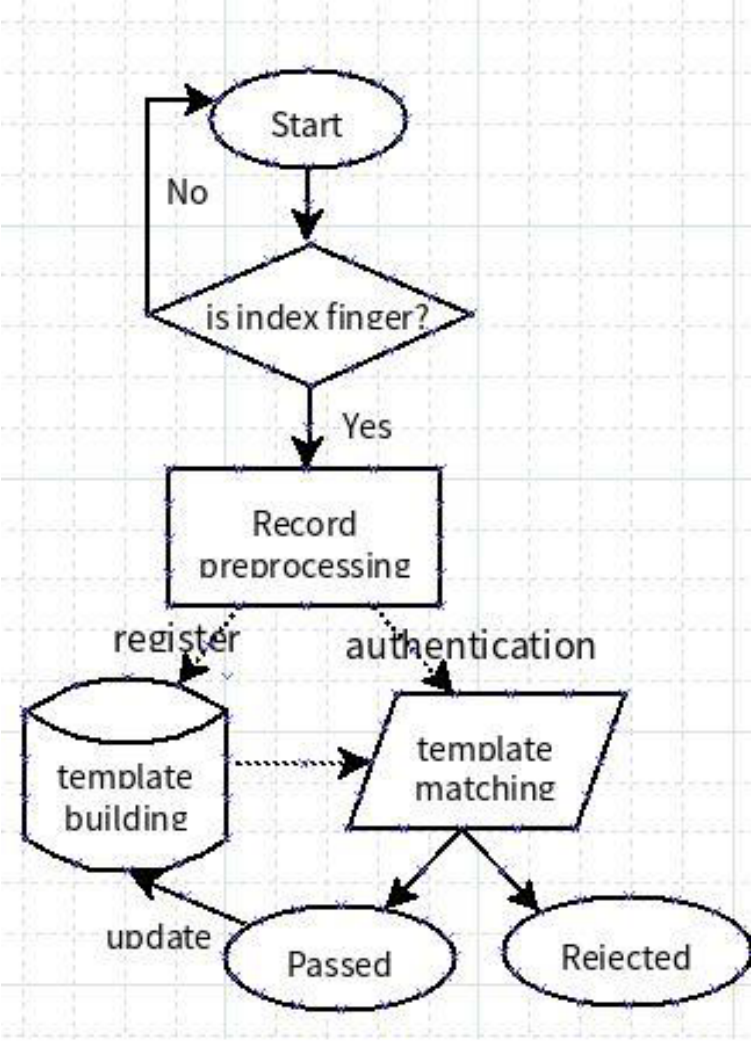


Figure 5.2 Process of System

5.2 Data Processing

5.2.1 Data Acquisition

Depth camera is a kind of balance of traditional methods above, on the one hand, it has embedded RGB camera, which can capture the movement of hands, it also has infrared radiation sensor, which can eliminate errors caused by the external environment. On the other hands, it is user friendly and can be used as long as there is a computer.

After built the basic environment in our research, First to do is get the raw data of hand gesture. Leap Motion will return gesture information due to the what kind of code we wrote in the program, including palm or fingertips position , direction and speed of fingertips and so on. The basic gesture information will be record frame by frame, and shown in the below format:

$$\begin{aligned} frame = \{&frameid, timestamp, \\ &palmposition, fingertipposition, \\ &fingertipspeed, etc\} \end{aligned} \tag{5.1}$$

Where position or speed part consists of data from x, y and z axis. We did some programming on the raw sequence, and pick up each feature by extracting three axis data of this feature. If there are no hands over Leap Motion, the system will only return frame id and timestamp as figure 5.3

```
Frame id: 139109, timestamp: 3010624127390, hands: 0, fingers: 0
Frame id: 139110, timestamp: 3010624135990, hands: 0, fingers: 0
Frame id: 139111, timestamp: 3010624144738, hands: 0, fingers: 0
```

Figure 5.3 Frame data when no hands over camera

Once user put hands over the Leap Motion, camera will sense hands information immediately, then data of whole hand will return to computer as figure 5.4 shows

```
Frame id: 179151, timestamp: 3010977233497, hands: 1, fingers: 5
  Right hand, id: 5, palm position: (-215.943, 108.402, 122.076)
  pitch: 7.854264191708504 degrees, roll: -21.893066601377623 degrees, yaw: 11.853199078099339 degrees
  Arm direction: (-0.715418, 0.0566897, -0.696393), wrist position: (-218.239, 102.581, 172.131), elbow position: (-40.1247, 88.4674, 345.509)
    TYPE_THUMB, id: 50, length: 48.056133mm, width: 18.67238mm
      TYPE_METACARPAL bone, start: (-246.397, 103.501, 169.601), end: (-246.397, 103.501, 169.601), direction: (0, 0, 0)
      TYPE_PROXIMAL bone, start: (-246.397, 103.501, 169.601), end: (-272.607, 107.212, 131.946), direction: (0.569425, -0.0806291, 0.81808)
      TYPE_INTERMEDIATE bone, start: (-272.607, 107.212, 131.946), end: (-284.524, 103.607, 103.077), direction: (0.379032, 0.114668, 0.918252)
      TYPE_DISTAL bone, start: (-284.524, 103.607, 103.077), end: (-289.031, 97.9977, 82.7318), direction: (0.208886, 0.259915, 0.942768)
    TYPE_INDEX, id: 51, length: 54.225983mm, width: 17.835857mm
      TYPE_METACARPAL bone, start: (-231.049, 118.266, 164.686), end: (-231.494, 122.422, 96.9772), direction: (0.00655963, -0.0612681, 0.9981)
      TYPE_PROXIMAL bone, start: (-231.494, 122.422, 96.9772), end: (-234.777, 122.769, 57.4997), direction: (0.0828871, -0.00875231, 0.996521)
      TYPE_INTERMEDIATE bone, start: (-234.777, 122.769, 57.4997), end: (-239.542, 115.112, 37.1183), direction: (0.213785, 0.343543, 0.91448)
      TYPE_DISTAL bone, start: (-239.542, 115.112, 37.1183), end: (-244.828, 103.722, 27.603), direction: (0.33551, 0.722944, 0.603974)
    TYPE_MIDDLE, id: 52, length: 61.786156mm, width: 17.517181mm
      TYPE_METACARPAL bone, start: (-220.032, 115.631, 163.908), end: (-212.297, 116.218, 100.045), direction: (-0.120227, -0.0091185, 0.992705)
      TYPE_PROXIMAL bone, start: (-212.297, 116.218, 100.045), end: (-207.797, 115.467, 55.8343), direction: (-0.101263, 0.0168829, 0.994716)
      TYPE_INTERMEDIATE bone, start: (-207.797, 115.467, 55.8343), end: (-211.058, 105.838, 31.6646), direction: (0.124365, 0.367253, 0.921769)
      TYPE_DISTAL bone, start: (-211.058, 105.838, 31.6646), end: (-217.744, 93.2742, 21.7793), direction: (0.385857, 0.72503, 0.570479)
    TYPE_RING, id: 53, length: 59.40904mm, width: 16.668709mm
      TYPE_METACARPAL bone, start: (-209.978, 110.427, 164.316), end: (-195.202, 107.455, 108.557), direction: (-0.255819, 0.0514495, 0.965355)
      TYPE_PROXIMAL bone, start: (-195.202, 107.455, 108.557), end: (-184.66, 104.294, 68.8555), direction: (-0.25588, 0.076744, 0.963657)
      TYPE_INTERMEDIATE bone, start: (-184.66, 104.294, 68.8555), end: (-184.67, 94.1503, 45.412), direction: (0.000404411, 0.397099, 0.917776)
      TYPE_DISTAL bone, start: (-184.67, 94.1503, 45.412), end: (-190.361, 81.9507, 34.6602), direction: (0.33032, 0.70811, 0.624074)
    TYPE_PINKY, id: 54, length: 46.57559mm, width: 14.80645mm
      TYPE_METACARPAL bone, start: (-202.98, 99.5438, 166.507), end: (-182.003, 95.7866, 117.47), direction: (-0.39233, 0.0702703, 0.917136)
      TYPE_PROXIMAL bone, start: (-182.003, 95.7866, 117.47), end: (-166.604, 90.0568, 89.308), direction: (-0.472279, 0.175738, 0.863753)
      TYPE_INTERMEDIATE bone, start: (-166.604, 90.0568, 89.308), end: (-163.271, 82.2892, 73.3765), direction: (-0.184853, 0.430693, 0.883365)
      TYPE_DISTAL bone, start: (-163.271, 82.2892, 73.3765), end: (-167.166, 71.6269, 62.2517), direction: (0.245079, 0.670841, 0.699935)
```

Figure 5.4 Frame data of one hand

For each feature, we extracted data and make them in time series as belows:

$$f = \{(x_1^k, y_1^k, z_1^k), ..., (x_n^k, y_n^k, z_n^k)\} \tag{5.2}$$

Here, n is the number of frames, and each tuple(x, y, z) is one feature data in one frame. After that we will get the raw data of each gesture. Figure5.5 give the example of raw feature sequence consist of the location of fingertip. Figure 5.6 shows the raw location coordinate data of x,y and z axis.

```
-1.316285,-0.495234,-0.846866<br>-1.135068,-0.540829,-0.897220<br>-0.942268,-0.581317,-0.946003<br>
```

Figure 5.5 location of one fingertip

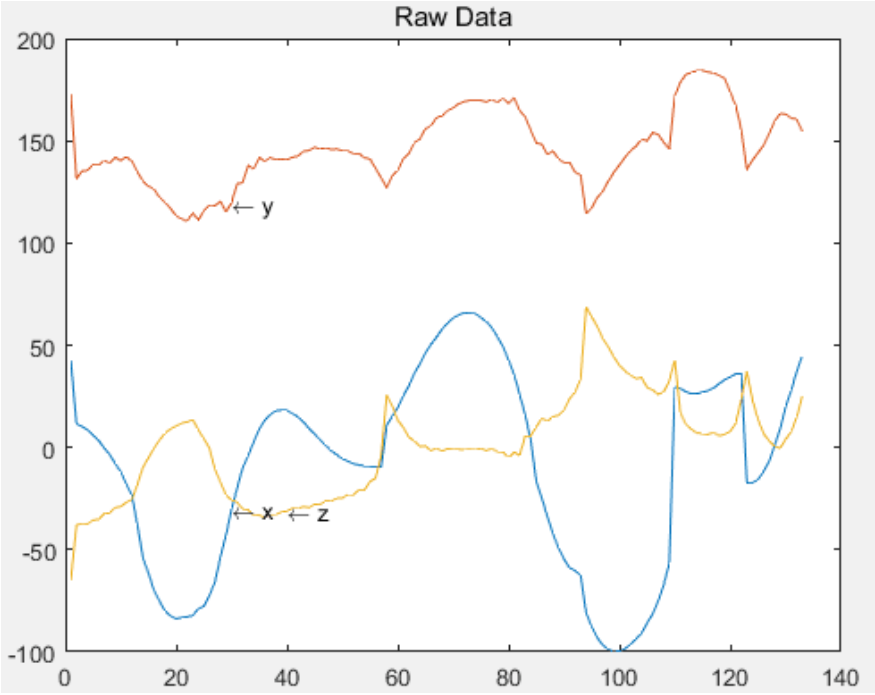


Figure 5.6 Raw data

We used x axis data of fingertip location as a example to introduce how we process with the raw data. The raw x axis data is in following format

$$l_x = (x_1, x_2, ..., x_i, ..., x_n) \tag{5.3}$$

5.2.2 Data Smooth

As chapter 4 shows, Kalman filter can smooth current data based on previous state, in (5.3), each x_i is a x axis location state of time t_i.

We used formula (4.3) - (4.7) to smooth each state of one sequence. For each x in sequence,

1. First, we used 4.3 to estimate current $x_{t|t-1}$ value based on previous optimal x_{t-1} Since there are no external control in our system, the parameter B can be 0, and parameter A, H in (4.1) and (4.2) can be treated as 1;

2. Then, we used 4.4 to calculate the covariance of estimated x_t;

3. Next, we used 4.6 to calculate the Kalman Gain of the current time;

4. Then, we used 4.5 to get the current optimal x_t by estimated value $x_{t|t-1}$ and observed value y_t;

5. Finally, we updated the covariance by 4.7;

We calculate each sequence by algorithm above, then, the data format for three axis of a feature will be smoothed sequence

$$s^k = \{(sx_1, sy_1, sz_1), ..., (sx_n, sy_n, sz_n)\} \tag{5.4}$$

we can get smooth result as following figure 5.7 shows

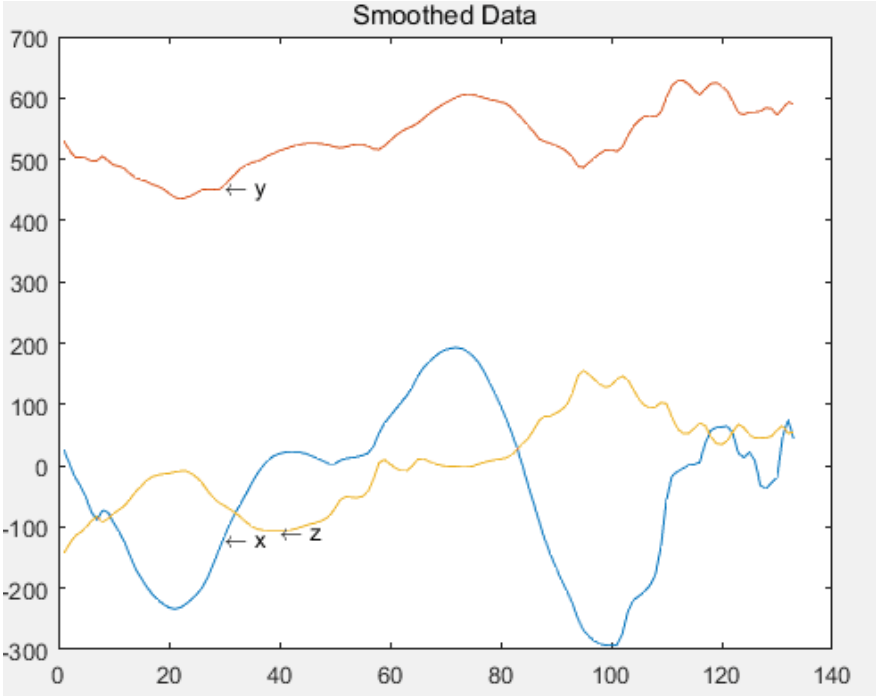


Figure 5.7 Smoothed data

5.2.3 Data Normalization

For each sequence of three axis, we calculated the standard score and got the normalized sequence. After that, the smoothed data will be changed into normalized sequence

$$n(x,y,z) = \{(nx_1, ny_1, nz_1), .., (nx_n, ny_n, nz_n)\} \tag{5.5}$$

The normalized data is shown in Figure 5.8.

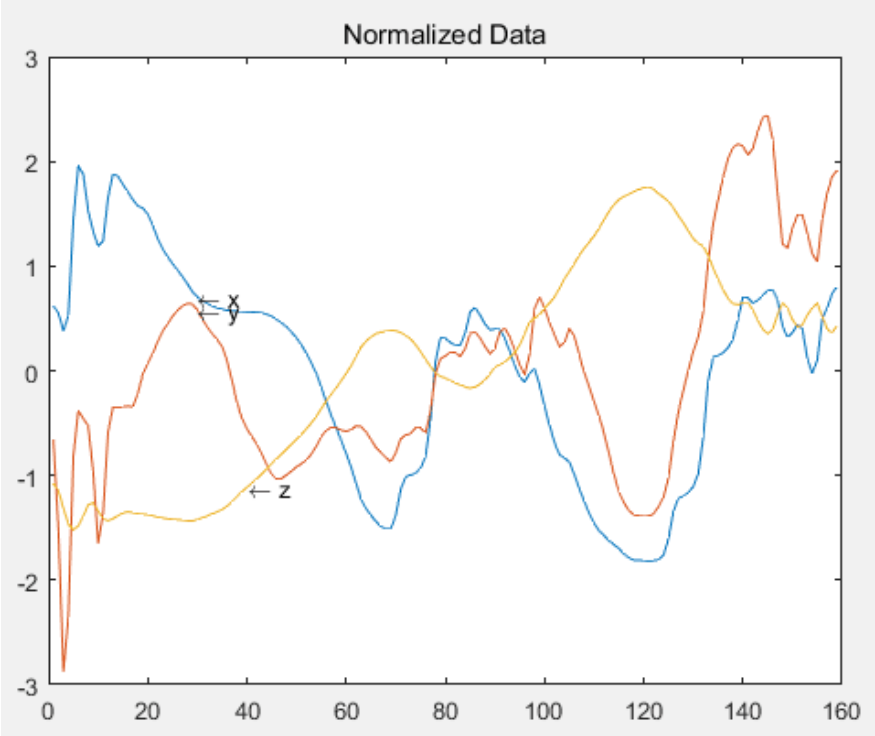


Figure 5.8 Normalized data

5.3 Feature Extraction

In order to collect suitable features, a kind of cluster method is used in our system.

First, we put all features that given by Leap Motion into a feature set X, including speed, track, and other informations of fingertip and palm and bones.

$$X = \{F_1, F_2, ..., F_n\} \tag{5.6}$$

Here, F represents one kind of features such as Fingertip location, speed and so on. Then, we build a empty set Y to store optimal feature combination. The feature extraction steps are

1. Test each feature of X and find the feature F_i that performed best in X, put it into optimal set Y, and remove it from X.

2. Then, for remaining features in X, pick up one feature F_i each time and combine it with features of Y, test the new set and get result. if the performance is better than using current optimal set Y, the new set will replace Y set, otherwise Y will not be changed and F_i will be discarded.

3. Repeat step 2 until feature set X become empty.

After that, optimal set Y will be the appropriate feature combination and will be used in classification.

For example, at first we have lots of features as figure 5.4 shows, and we first choose each feature independently as initial feature. and test the accuracy as follow table 5.1 shows

Table 5.1 Accuracy of using different features

Feature	palm center position	fingertip position	fingertip speed	wrist position
Accuracy	86.2%	88.4%	84.3%	83.2%
Feature	metacarpal bone	proximal bone	intermediate bone	distal bone
Accuracy	84.5%	81.6%	84.3%	81.2%

We can see the position of fingertip perform best above all features, we put it in to the optimal set Y at first, and remove it from feature set X. Then for features remain in X, we picked up one feature each time and made the union for Y and this feature, testing the accuracy and got the result as figure 5.9 shows.

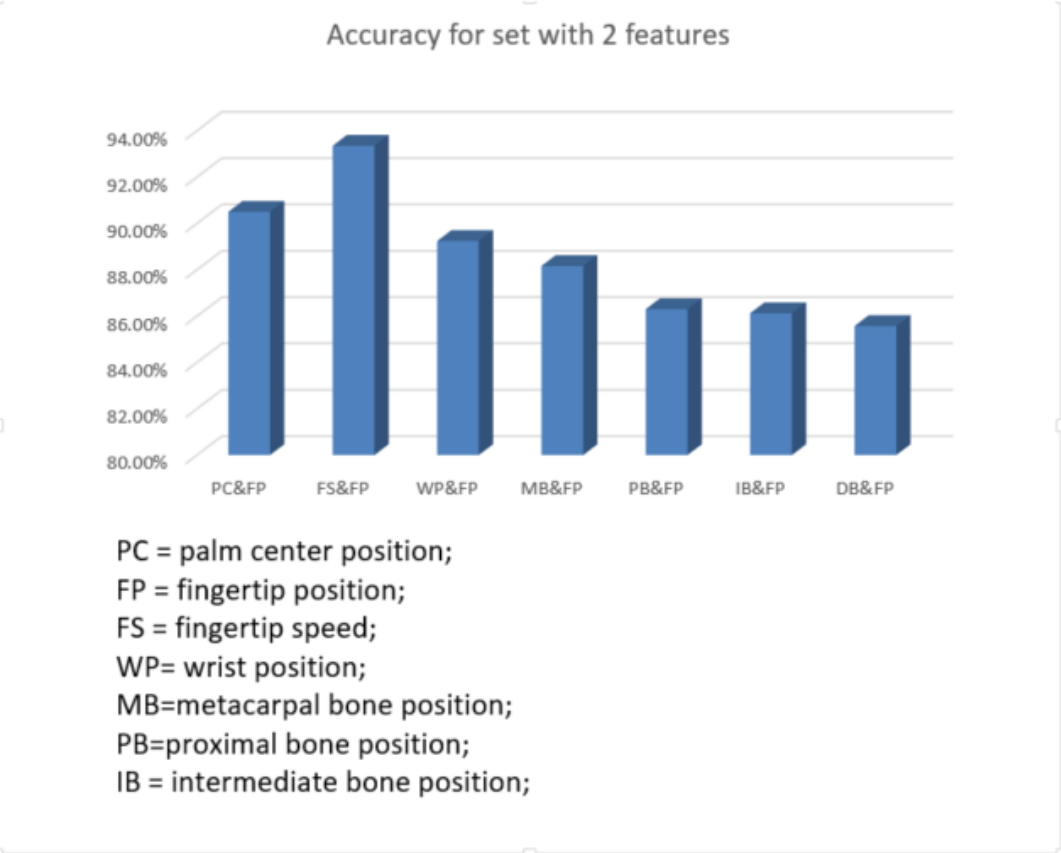


Figure 5.9 Normalized data

Therefore, the best combination of 2 features is fingertip position and fingertip speed. Since the accuracy of feature set with 3 features are not better than accuracy of fingertip position and speed, we decided to use the combination of this two features.

5.4 Classification

We used trajectory and speed as our features to be used in classifier.

1. First, we make the gesture information for each timestamp in following format

$$f^k = (p_x^k, p_y^k, p_z^k, v_x^k, v_y^k, v_z^k) \tag{5.7}$$

while p is the position of one fingertip, and v is the speed of this fingertip, and f^k is one state of this gesture sequence in time k.

2. Then we put the sequence into the classifier, and use Algorithm 2 to build the model for each features.

The classifier will works when f^k is inputed. First it calculate the state transition probability for each f^k, then we used the result of training set and calculate the output probability for each state. By calculating probabilities for each state, the classifier algorithm will build a whole hidden markov model for gesture data.

5.5 Template Updating

We used a double threshold method to update our gesture template, one threshold is the successfully accessed times, another is the build time for a template. In our research, the template updating mechanism work in following sequence.

1. First, the system will build the template for each user and record the last access date and access times.

2. If a user's accepted times achieved the threshold, the new accepted gesture will be attached to the existed template.

3. If the access time does not achieved the threshold, but it has been a long time after template built or last update times, the system will also update new gesture to the template.

4. Finally, system will update timestamp and access times to be used in next time.

Chapter 6

Experiments

We mainly did some experiments with the accuracy of original system and template updating mechanism.

6.1 Evaluation Standard

Before experiment, we will first introduce our evaluation standard.

To evaluate an authentication system, first to think is the False Acceptance Rate (FAR) because the performance usually depends on the ability of defending. It also should keep good performance on False Rejection Rate(FRR).

False Acceptance Rate is the probability that the system accepted a non-authorized person, False Rejection Rate is the probability that the system incorrectly rejects an genuine person. Generally, in an authentication system, the two rate are conflict with each other as Figure 6.1 shows

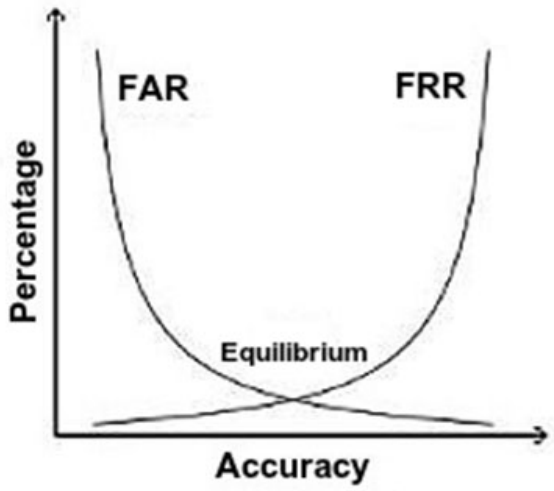


Figure 6.1 FAR and FRR equilibrium

In an identification system, false rejection rate is always concerned more important and should be decreased, but in a verification system, in order to have high security performance, false acceptance rate should be emphasized to prevent from attacks, in such case, false rejection rate can be relative high to reduce false acceptance rate.

6.2 Accuracy of General Authentication

In this part, we invited 4 users aged 22-24, each of the user have no experience with such kind of system but have related knowledge about how to use Leap Motion. The experiment step are as follows

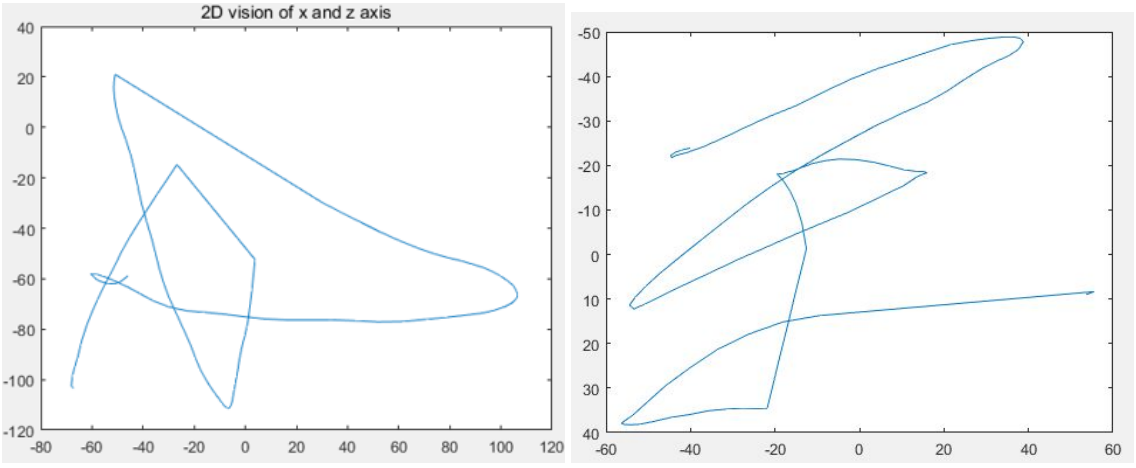
1. Each user registers a gesture and perform it 20 times, then the system will store the gesture as their template. And they can see the gesture of other users.

2. After built the template, each user try to pass the authentication with their own hand gesture for 20 times, this step will give the false rejection rate of the system.

3. Then, each user try to imitate each others gesture and try to attack their "account", then the false acceptance rate will be recorded.

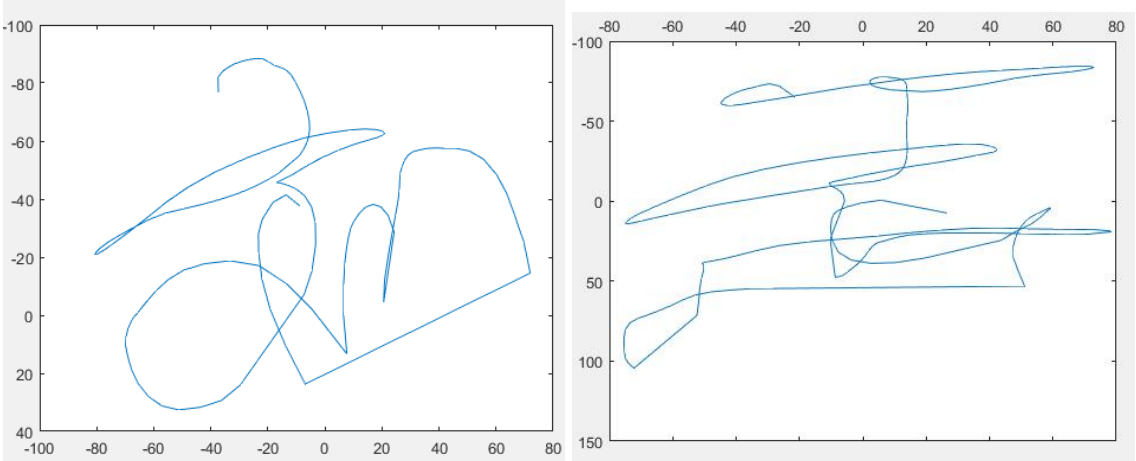
4. Finally, calculate the total error using all data from step 2 and 3.

We did experiments with simple and complicated gesture separately, following figures show some examples of two kinds of gesture in 2D vision. Simple gestures are gestures like drawing a star or other symbol with little strokes, complicated gestures means symbol with lots of strokes. All of the symbols are drawn without interrupt.



(a) Simple (b) Simple

Figure 6.2 Simple gesture



(a) Complicated (b) Complicated

Figure 6.3 Complicated gesture

Then we test the system accuracy with simple and complicated gesture, The FAR and FRR result of simple gesture group is as figure 6.4 shows

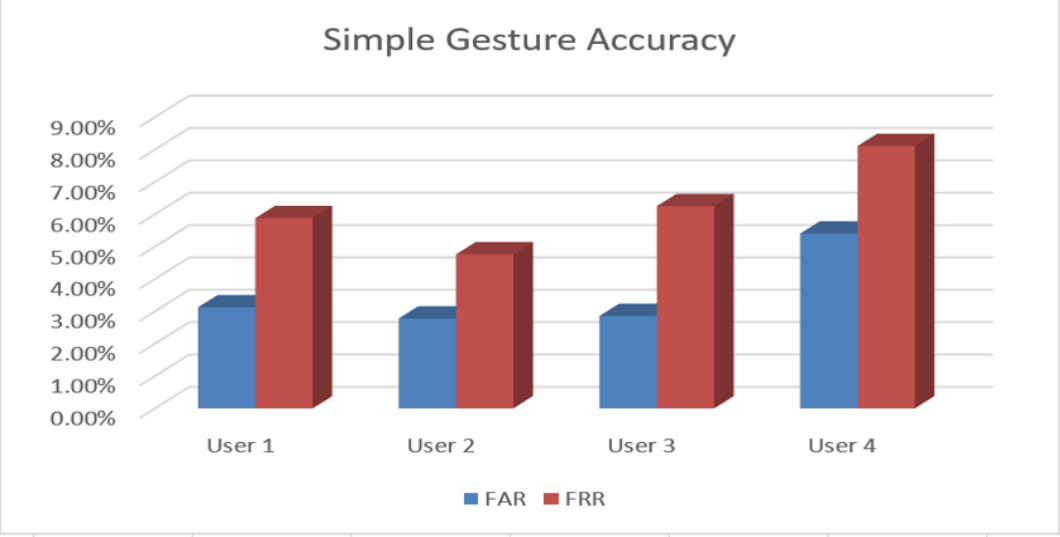


Figure 6.4 Accuracy result of Simple Gesture

And the FAR and FRR of complicated gesture is as figure 6.5 shows.

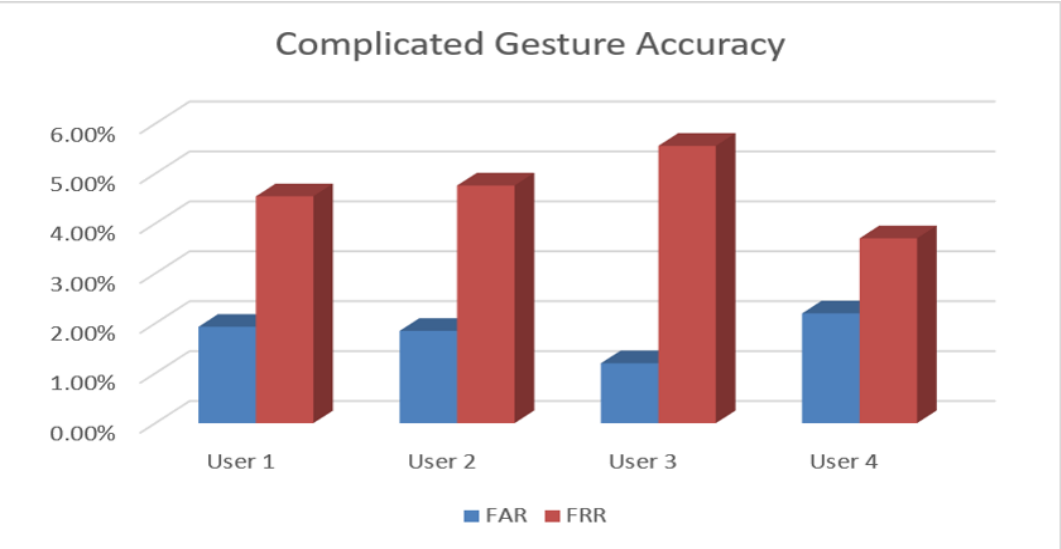


Figure 6.5 Accuracy Result of Complicated Gesture

As figure 6.4 shows, in the case of simple gesture, the average accuracy is 91.38%, while false acceptance rate is 3.62%, false rejection rate is 6.57%. And in the case of complicated gesture as figure 6.5, the average accuracy is 95.21%, with 1.65% false acceptance rate and 4.82% false rejection rate.

The accuracy is listed in following table 6.1

Table 6.1 Accuracy of Simple and Complicated Gesture

Group	**Average FAR**	**Average FRR**	**Accuracy**
Simple gesture	3.62%	6.57%	91.38%
Complicated gesture	1.65%	4.82%	95.21%

6.3 Accuracy of Template Updating Mechanism

In the experiments of template updating mechanism, we designed steps as follows,

1. First, we invited 10 participants (aged 21-24) who are also not familiar with this system but

have experience of using Leap Motion to join the experiment.

2. Then, 10 users are asked to perform their gesture they like, which will be registered as their gesture password.

3. After 3 days, 10 users were asked to access their account with registered gesture, 5 of the users used system with template updating mechanism(Group T), so their gesture will be put into template record. Others(Group O) used general system without updating template. And then we recorded the accuracy with FAR and FRR.

4. After one week and another week, they were asked to access their account again as step 2 works, and we record the accuracy.

5. Finally, we compared three accuracy record and get the conclusion.

The accuracy of template updating mechanism are shown in the following figures 6.6.

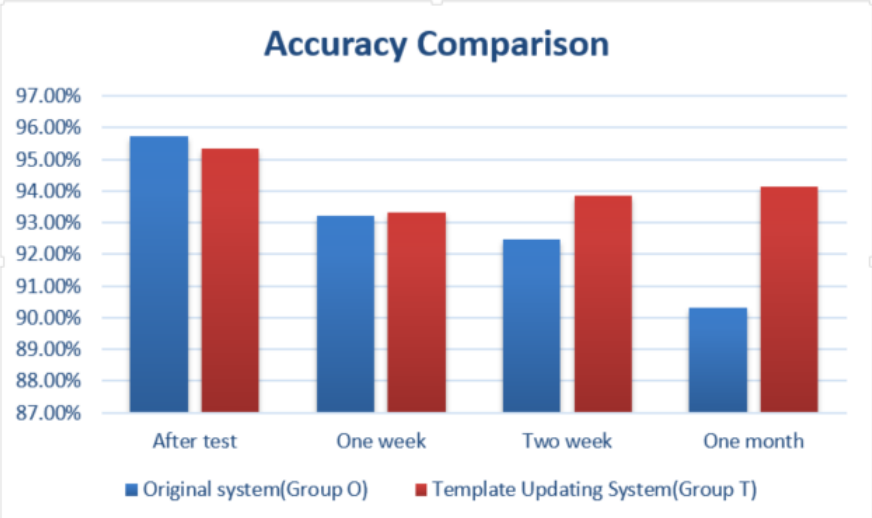


Figure 6.6 Accuracy Result with Template Updating

We can see, at first, two group have almost same accuracy(T = 95.34%, O = 95.72%) after one week, the accuracy of two group are generally equal and did not changed a lot, but after two week

and one month, the accuracy of group O dropped, while group T keeps a relatively high accuracy. Hence, we can prove that our template updating mechanism works in this system.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In the thesis, we generated a dynamic hand gesture authentication method by using Leap Motion. User can finish the authentication by performing different gesture over Leap Motion. This thesis also come up with a series of data processing. feature extraction and classification for gesture data. From the experiments, the system works with a relatively high accuracy.

From the general accuracy result we can see our system got an over 95% accuracy with 1.65% false acceptance rate with complicated dynamic hand gesture, compared with previous research[22], we made some improvement on the accuracy performance. Besides, we introduced template updating mechanism of our system, by using this mechanism, the system have keep its authentication accuracy with time passing, which proved the good permanence and robust of our system. This is also similar to text password we used in real life, complicated password perform better on security, but remember gesture is much simpler than remember long text password, also ,gesture can not be lost, and not easy to be copied.

From the experiment result, it is obvious that complicated gesture perform better than simple gesture, which has same reason with current text password but gesture is easier to remember.

Hence, we recommend to use a complicated gesture such as personal sign in our system.

After the experiments, most of participants think our system is easy to use and have better security performance than traditional password, which proved the usability of our methods.

7.2 Future Work

Our research is aimed to use hand gesture to finish authentication. We mainly make some progress on data acquisition, processing, feature extraction and classification, but there are still exist some aspects that can be improved.

1. Build diversified hand gesture model In our research, we only used one finger and the dynamic gesture. In future work, the combination of static features and dynamic features may work better in getting high accuracy. One idea is use multiple fingers as a basis for identification.

2. Simplify the registration process In our research, we required every user perform over 20 times to build their own template, which is not convenient in practice, what we should do next is to find a method that can reduce the repetition while keep the accuracy and robust of template.

3. Propose a more reliable template update algorithm In our design, we used double threshold mechanism which consists of time and template length to reduce the influence of false acceptance situation and improve the permanence. But such kind of methods can not completely eliminate the threaten of false acceptance. We will think about how to improve the template updating mechanism to make the system safer.

Biometric authentication has great potential because of its uniqueness and universality, gesture authentication is only one type of biometric authentication. To combine gesture and other biometric information to make a stable and accurate authentication system, or to make it replace the

traditional text password, we still need a lots of thinking and experiments. That will also be the direction of our future efforts.

References

[1] W. Melicher, D. Kurilova, S. M. Segreti, P. Kalvani, R. Shay, B. Ur, L. Bauer, N. Christin, L. F. Cranor, and M. L. Mazurek, "Usability and security of text passwords on mobile devices," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 527–539.

[2] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *Woot*, vol. 10, pp. 1–7, 2010.

[3] A. H. Lashkari, S. Farmand, D. Zakaria, O. Bin, D. Saleh *et al.*, "Shoulder surfing attack in graphical password authentication," *arXiv preprint arXiv:0912.0951*, 2009.

[4] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[5] J. Daugman, "How iris recognition works," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 21–30, 2004.

[6] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[7] Y. Chen, J. Sun, R. Zhang, and Y. Zhang, "Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2686–2694.

[8] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor." *JCP*, vol. 1, no. 7, pp. 51–59, 2006.

[9] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[10] S. Kratz and M. T. I. Aumi, "Airauth: a biometric authentication system using in-air hand gestures," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014, pp. 499–502.

[11] A. Chahar, S. Yadav, I. Nigam, R. Singh, and M. Vatsa, "A leap password based verification system," in *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 1–6.

[12] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.

[13] M. T. I. Aumi and S. Kratz, "Airauth: evaluating in-air hand gestures for authentication," in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 2014, pp. 309–318.

[14] Y. Yang, G. D. Clark, J. Lindqvist, and A. Oulasvirta, "Free-form gesture authentication in the wild," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 3722–3735.

[15] S. Kamaishi and R. Uda, "Biometric authentication by handwriting using leap motion," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*. ACM, 2016, p. 36.

[16] Y. Li, "Protractor: a fast and accurate gesture recognizer," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 2169–2172.

[17] Z. Sun, Y. Wang, G. Qu, and Z. Zhou, "A 3-d hand gesture signature based biometric authentication system for smartphones," *Security and Communication Networks*, 2016.

[18] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]," *IEEE Signal processing magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[19] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[20] M. F. Nowlan, "Human identification via gait recognition using accelerometer gyro forces," *Yale Computer Science. http://www. cs. yale. edu/homes/mfn3/pub/mfn_gait_id. pdf (accessed November 12, 2013)*, 2009.

[21] Z. Zhou, *Machine Learning*. Tsinghua University Press, 2016.

[22] 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.

Appendix A

Appendix

A.1 How to use our system

Our system are built under Java JDK 1.8 and windows 10 operating system. First the computer should install this kinds of environment. Then, users should connect Leap Motion with computer throw USB port and cable as Figure A.1.

Figure A.1 Connect Leap Motion to Computer

One caution is that the green indicator light should be directly facing to the user, otherwise the coordinate will lost the accuracy.

Then users can run our system by Java, and the interface of system is shown in figure A.2

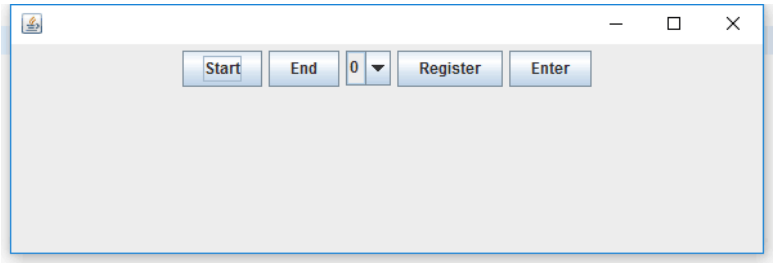


Figure A.2 Interface of System

The user can select his account, in order to facilitate the experiment, we used drop down box to represent the different users in figureA.3

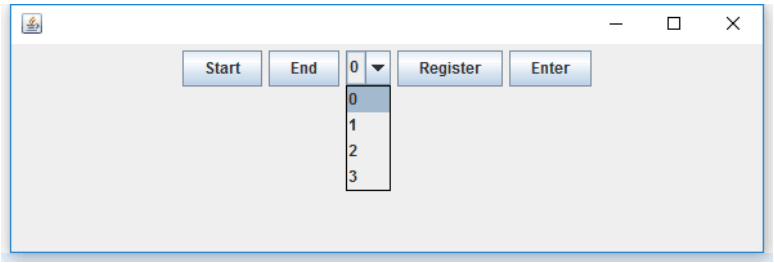


Figure A.3 Select user account

After chosen the user account, user should click "start" button and perform the gesture like figure A.4 shows

Figure A.4 Perform Gesture

In register part, users can store their gesture as template data by clicking "Register" button, the input gesture will be automatically stored into the template files. In access part, user can try to access the system by click "Enter" button, then the system will give the feedback that whether current user passed or not like figure A.5 and A.6 shows.

Figure A.5 Passed

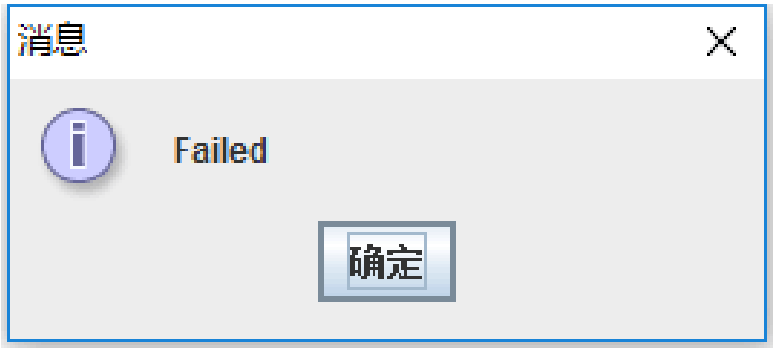


Figure A.6 Failed to pass