



澳門大學  
UNIVERSIDADE DE MACAU  
UNIVERSITY OF MACAU

# Outstanding Academic Papers by Students

## 學生優秀作品



**University of Macau**

**Faculty of Science and Technology**



**澳門大學**

**UNIVERSIDADE DE MACAU**

**UNIVERSITY OF MACAU**

# **A Cost-effective Palm Vein Recognition System**

*by*

**Zhu Licheng, Student No: DB327011**

Final Project Report submitted in partial fulfilment  
of the requirements of the Degree of  
Bachelor of Science in Computer Science

Project Supervisor

Prof. Zhang Yibo

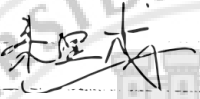
16 May 2017

## DECLARATION

I sincerely declare that:

1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature :



Name :

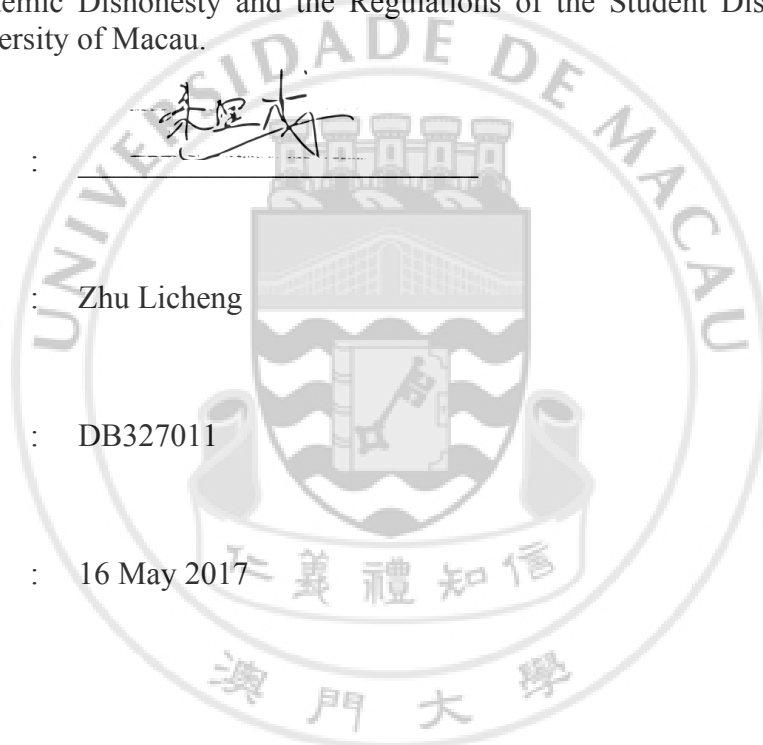
Zhu Licheng

Student No. :

DB327011

Date :

16 May 2017



## ACKNOWLEDGEMENTS

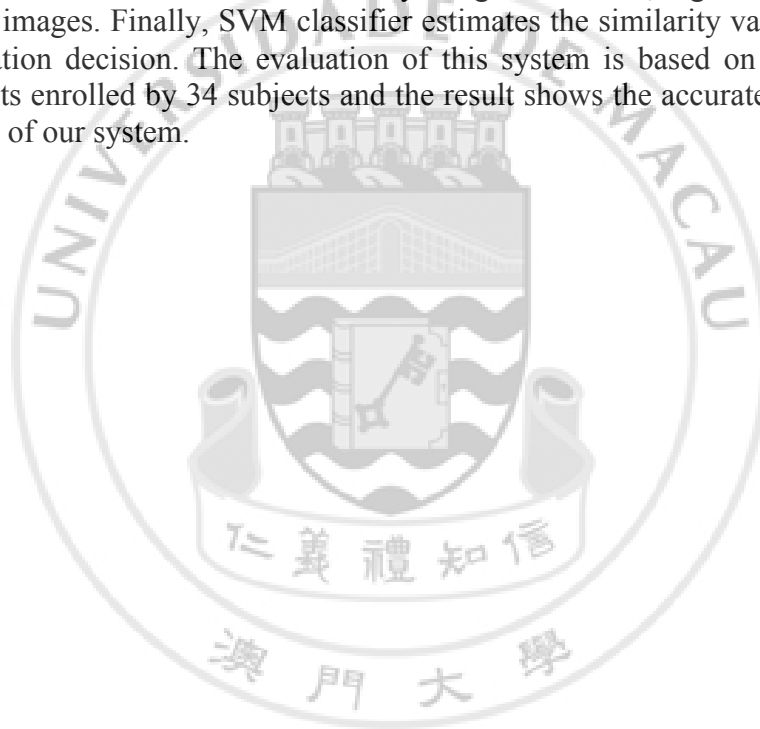
The author would like to express his utmost gratitude to UM for providing the opportunity to carry out a project as a partial fulfilment of the requirement for the degree of Bachelor of Science.

Throughout this project, the author was very fortunate to receive the guidance and encouragement from his supervisor Zhang Yibo.



## ABSTRACT

Nowadays, traditional identity verification methods like typing in passwords have been gradually replaced by verification based on your physiological characteristics or behavioural patterns, known as biometrics because of their advantages in security and reliability. Palm vein recognition is outstanding compared to other techniques because of the difficulty in circumvention. However, the existing palm vein recognition devices on the market are extremely expensive. In this project, we developed the software based on MFC framework and MATLAB with a low-cost sensor module produced by a 3D printer. However, we still achieve a very high precision in recognition. Multi-scale and multi-direction Gaussian filter is first applied to extract the palm vein of ROI. Then the image is transformed into a binary image and small noisy regions are eliminated from the binary image. After that, registration is applied to align two images. Finally, SVM classifier estimates the similarity value and makes an identification decision. The evaluation of this system is based on a database of 1156 data sets enrolled by 34 subjects and the result shows the accurate classification performance of our system.



## TABLE OF CONTENTS

<b>CHAPTER 1. INTRODUCTION.....</b>	<b>10</b>
1.1 BACKGROUND.....	10
1.2 OBJECTIVE.....	11
1.3 DATA SOURCE .....	11
1.4 DEVELOP ENVIRONMENT .....	11
1.5 SCHEDULE .....	11
<b>CHAPTER 2. RELATED WORK.....</b>	<b>13</b>
<b>CHAPTER 3. FUNCTIONAL SPECIFICATION.....</b>	<b>14</b>
3.1 USE CASE DIAGRAM .....	14
3.2 USER .....	14
3.2.1 Registration.....	14
3.2.2 Verification.....	14
3.2.3 Identification .....	14
3.3 MAINTENANCE.....	15
3.3.1 Maintain database .....	15
3.3.2 Update classifier.....	15
3.4 INSTALLATION.....	15
<b>CHAPTER 4. SOFTWARE DESIGN SPECIFICATION .....</b>	<b>16</b>
4.1 SYSTEM ARCHITECTURE.....	16
4.2 SEQUENCE DIAGRAM .....	16
4.2.1 Registration.....	17
4.2.2 Verification.....	18
4.2.3 Identification .....	19
4.3 Interface description .....	20
<b>CHAPTER 5. IMPLEMENTATION NARRATIVES AND DESCRIPTION .....</b>	<b>26</b>
5.1 IMAGE PROCESSING .....	26
5.1.1 ROI (Region of Interest).....	26
5.1.2 Gaussian Filter.....	26
5.1.3 Transform Grey Level to Binary.....	28
5.1.4 Eliminating Small Image Regions.....	28
5.1.5 Image Registration .....	29

5.2	Comparison .....	31
5.3	SVM Classifier .....	32
5.4	MFC Framework .....	35
5.5	Multiple Platform .....	35
<b>CHAPTER 6. SYSTEM QUALITY .....</b>		<b>36</b>
6.1	A Genuine Individual in Verification Mode .....	36
6.2	An imposter in Verification Mode .....	37
6.3	A Genuine Individual in Identification Mode .....	38
6.4	An Imposter in Identification Mode .....	38
<b>CHAPTER 7. ETHICS AND PROFESSIONAL CONDUCT .....</b>		<b>40</b>
7.1	Use of Data .....	40
7.2	Coding .....	40
7.3	Miscellaneous .....	40
<b>CHAPTER 8. SUMMARY .....</b>		<b>41</b>
<b>CHAPTER 9. REFERENCES .....</b>		<b>42</b>

## LIST OF FIGURES

Figure 1: Image Extraction Device .....	10
Figure 2 Use Case Diagram .....	14
Figure 3 System Architecture .....	16
Figure 4 Registration Sequence Diagram .....	17
Figure 5 Verification Sequence Diagram .....	18
Figure 6 Identification Sequence Diagram .....	19
Figure 7 Main Interface .....	20
Figure 8 Register Interface.....	21
Figure 9 User Name Warning.....	21
Figure 10 Capture Button.....	21
Figure 11 Verification Interface.....	22
Figure 12 Verify Button.....	22
Figure 13 Capture Complete Notification.....	23
Figure 14 Verification Success Notification.....	23
Figure 15 Verification Failure Notification .....	24
Figure 16 Identify Button.....	24
Figure 17 ROI of Original Image.....	26
Figure 18 Gaussian Function .....	27
Figure 19 Gaussian Filter Result with Different $\sigma$ .....	27
Figure 20 Gaussian Filter Result with Different Directions .....	28
Figure 21 Binary Transformation with Different Thresholds.....	28
Figure 22 Binary Palm Vein with Many Noisy Regions .....	29
Figure 23 Binary Palm Vein after Eliminating Small Regions.....	30
Figure 24 Binary Palm Vein before Registration .....	30



Figure 25 Registration Flow Chart .....	31
Figure 26 Binary Palm Vein Combination after Registration .....	31
Figure 27 SVM Classifier .....	33
Figure 28 Quadratic Kernel .....	35
Figure 29 RBF Kernel.....	35
Figure 30 Linear Kernel.....	36



## LIST OF TABLES

Table 1 Schedule.....	11
Table 2 SVM Kernel Comparison .....	34
Table 3 Test Case 1 .....	36
Table 4 Test Case 2.....	37
Table 5 Test Case 3.....	38
Table 6 Test Case 4.....	39



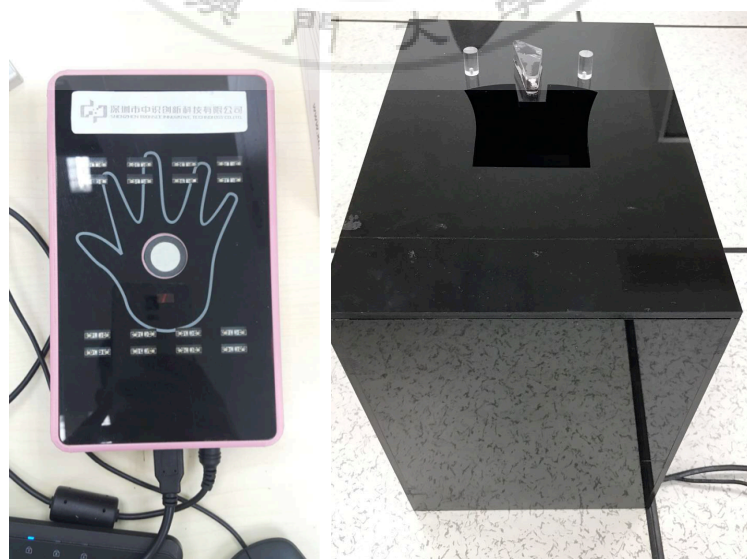
## CHAPTER 1. INTRODUCTION

### 1.1 BACKGROUND

No two people are the same and this is the basic theory for us to apply to the application of biometric verification. With this idea, we are able to make the use of various unique characteristics of human body like eyes, faces, voices and so on to enhance security and verification.

A palm vein recognition is a process of extracting the shape of veins from people's hands. Nowadays, fingerprint recognition is very famous in security door, mobile phone encryption, customer inspection and so on. As contrast, the real-time detection of fingerprint may encounter a lot of problems. For example, a wet hand may generate much noise causing the extracted fingerprint blurred and inexplicit. In addition, facial recognition device has strict requirement about user's standing posture and illumination effect. To solve this problem, we consider extracting people's vein instead of fingerprints.

Infrared is an invisible radiant energy with longer wavelength than those visible lights. Near-infrared emitted by our device can easily penetrate people's skin and reach the veins so that veins can be captured and used to recognize the identity of different people. One advantage of using palm vein Recognition is that it is most robust. The infrared would neglect any dirty stuff on the surface of palm and extract the vein accurately so as to improve the accuracy of recognition. Another advantage is that it has a high degree of circumvention and is more secure to recognize veins because it is under the skin and thus very hard to imitate the pattern. Normally, it would be quite easy for someone's fingerprint to be stolen, but when it comes to vein recognition, it is hard to imitate, at least impossible within a short period and that confirms the security of vein recognition. In this case, we design a device generated by 3D printer, which is very cheap (see Figure 1). In addition, we also ordered a supplementary box to fix user's hand and the device so that ROI can be extracted properly without to many captured image differences caused by motion difference.



*Figure 1: Image Extraction Device*

## 1.2 OBJECTIVE

Nowadays, palm vein recognizers on the market are very expensive to buy. The cheapest one I can find is up to 4300 HKD [1]. Our goal is to make good use of our very low cost device that costs only 2000 CNY (bought from Harbin Institute of Technology, Shenzhen) and achieve highly accurate palm vein recognition. The major challenges of our project are:

- (1) Find the region of interest (ROI).
- (2) Use Gaussian filter to smooth the captured image so that veins can be shown clearly on the image. It is important to do experiments and find proper parameters of the filter.
- (3) Apply registration algorithm to align two images in case of errors caused by position transition of hand and also remove noisy regions composed by small area.
- (4) Apply a threshold or classifier to convert the smooth result into a binary image so as to calculate the similarity of two images conveniently. After calculating the similarity of two images, we can then determine whether these two is a match.
- (4) Provide two modes for users to choose, verification and identification.
- (5) Collect a proper number of people's palm vein to construct our database for further testing and optimization.

## 1.3 DATA SOURCE

Our program utilizes people's palm veins to drive the recognition process, so it is of great significance to collect enough people's palm vein images. We contacted volunteers to come to our lab and help us collect their palm veins. The final size of our database is 34 people, containing 12 females.

## 1.4 DEVELOP ENVIRONMENT

In order to connect all the parts into a whole system, we propose an application interface constructed by C++ with the help of MFC framework. MFC is short for Microsoft foundation class library. It provides a lot of macros for windows message-handling which provides a more concrete way to edit codes without looking into the root of language because message-handling macros replaced the virtual function mechanism provided by C++. In addition, we also make good use of MATLAB to help us handle issues about image and return results to C++.

## 1.5 SCHEDULE

*Table 1 Schedule*

Duration	Tasks
7/2016 ~ 9/2016	<ul style="list-style-type: none"> <li>Requirement specification</li> </ul>

	<ul style="list-style-type: none"> <li>Interface design</li> </ul>
10/2016 ~ 11/2016	<ul style="list-style-type: none"> <li>Design a prototype of image processing algorithm</li> <li>Collect raw data in order to improve the algorithm</li> </ul>
12/2016	<ul style="list-style-type: none"> <li>Require the manufacturer to add a cover to our device to fix user's hand</li> </ul>
1/2017 ~ 3/2017	<ul style="list-style-type: none"> <li>Enhance the algorithm</li> <li>Combine interface written in C++ with Image processing code written in MATLAB</li> </ul>
3/2017 ~ 4/2017	<ul style="list-style-type: none"> <li>Collect people's palm veins to construct our database</li> </ul>
4/2017	<ul style="list-style-type: none"> <li>Based on the data we collected, adjust our algorithm to achieve a better precision in future</li> <li>Integrate the whole project</li> </ul>
5/2017	<ul style="list-style-type: none"> <li>Report writing</li> </ul>

## CHAPTER 2. RELATED WORK

Hong Kong PolyU developed a palm print authentication system. The system, however, costs a lot since it has to keep the device stable and robust for the sake of instability of fingerprint. Our device is much cheaper with a much smaller body size so it would save a lot of expense.

Kono et al. [1] demonstrated a personal identification system that is based on near-infrared finger vein patterns with a CCD camera. The enhanced finger vein pattern enabled 678 persons to be successfully identified. Wang et al. [2] gives fair comparisons of shape and texture based methods for vein recognition. A database of 500 images for 100 people is used to evaluate three methods and the Gabor feature gets the best performance with 80% recognition rates. Mirmohamadsadeghi et al. [3] proposed a promising new approach based on local texture patterns. Yang et al. [4] presents a novel finger vein recognition method base on a PBBM and the result is highly accurate, robust and reliable. Some relevant work to our task is by Rosdi et al. [5]. They propose to use the local line binary pattern (LLBP) as feature extraction technique. The equal error rate of their result is significantly lower than LBP and LDP. Moreover, they used 2-D Gaussian high-pass filter to pre-process the captured images. However, since veins always stretch close to a straight line, we consider it better to use inversed 1-D Gaussian filters with various directions so that the vein can be clearly captured and enhanced.

There are researchers doing the same research on palm vein and having achieved good result in this area. However, most of their devices are very expensive and hard to be put into industry. Ming et al. [6] proposed a vein enhancement method combining KL transform and hue-preserving colour image enhancement technology, which is robust to shadow effect. Gupta et al. [7] proposed the palm vein recognition based on hand dorsal image. Cao et al. [8] proposed a mobile identification for wide age range.

## CHAPTER 3. Functional Specification

### 3.1 USE CASE DIAGRAM



*Figure 2 Use Case Diagram*

Figure 2 is the use case diagram of my recognition system. Generally, two users are provided here with their functions corresponding to them respectively. This system is closely related with the database behind it, so the maintenance should take the responsibility to update or delete data, update classifier regularly to keep the system running smoothly for users.

### 3.2 USER

#### 3.2.1 Registration

When the user would like to log in this system for the first time, he has to go through the registration step first. The user needs to provide his username and palm vein to the system in this function. After that, the user's information would be transferred to the database for future verification if the information provided is valid.

#### 3.2.2 Verification

If a user already has registered before, he can use verification function to log in the system. In this function, the user needs to provide his username and palm vein to the system and it will check whether there is matching username and palm vein in the database. The result would be shown on the screen with text and colours.

#### 3.2.3 Identification

Identification generally costs a lot of time since it requires comparing with user's current palm vein directly with ones in the database one by one until a correct one is found or no one is found and end in error. However, it is a very convenient function since user just needs to provide his palm vein in this mode. The result would also be shown on the screen in the same way as verification.

### **3.3 MAINTENANCE**

#### **3.3.1 Maintain database**

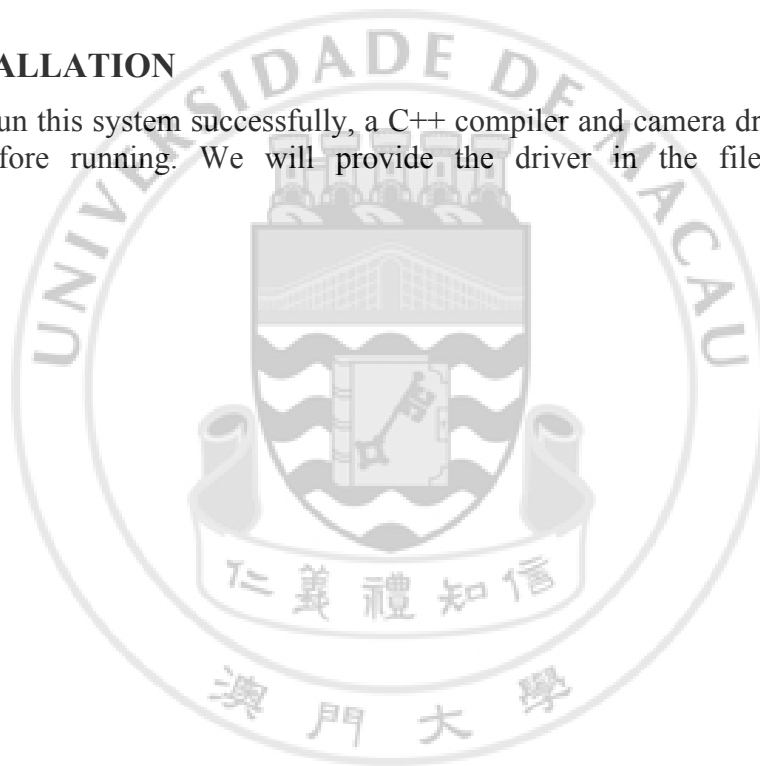
In this function, the maintenance is responsible for checking if there are useless data in the database and delete them; putting data in order when the size of data becomes too large and limits the running speed of the program. Generally, our system interacts closely with the database, so this function should be conducted well by the maintenance.

#### **3.3.2 Update classifier**

The classifier is used to figure out whether the newly input image has a match in our database or not. The system should be updating itself while being used. So the maintenance should update the classifier when a certain amount of data is added to our database to improve the performance of classification in the future.

### **3.4 INSTALLATION**

In order to run this system successfully, a C++ compiler and camera driver should be installed before running. We will provide the driver in the file package for installation.





## CHAPTER 4. Software Design Specification

### 4.1 SYSTEM ARCHITECTURE

Figure 3 shows the architecture of our system in detail. Sensor Module is used for capturing palm vein images and it is connected to PC through a wired line accompanied by specially designed driver. After the Sensor Module successfully captures the palm vein, the image would either be transferred to our database or Comparison Module for further process. The database is responsible for storing all the registered palm vein images. Comparison Module is the core module in this system since it takes charge of comparing captured palm and registered palm images and returning the result to Interface Module. Finally, the Interface Module is prepared to give a feedback to our users about the result of the recognition.

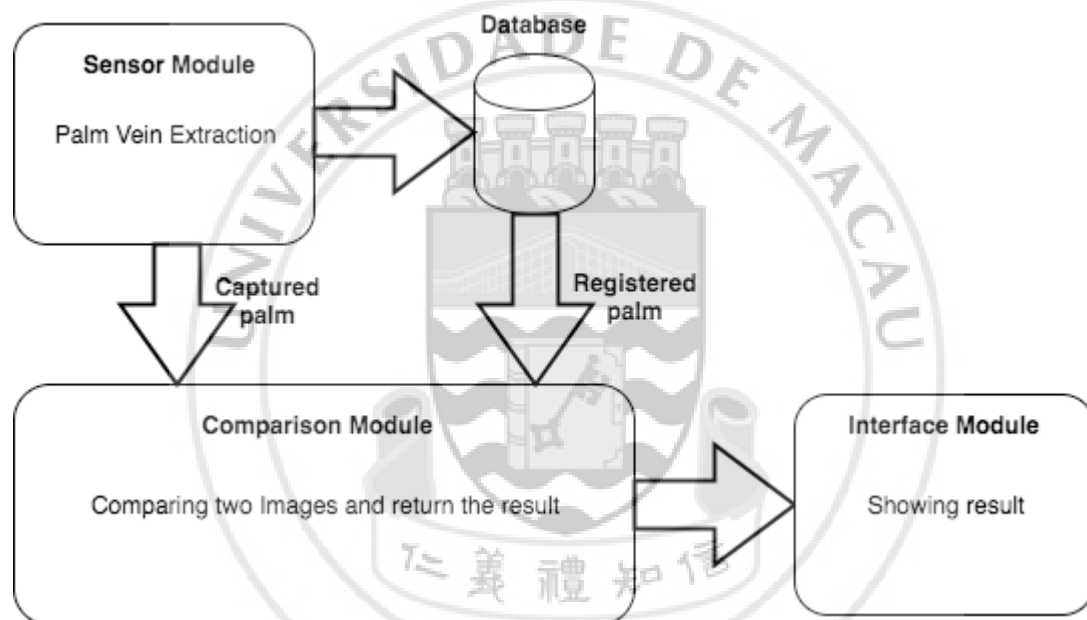


Figure 3. System Architecture

### 4.2 SEQUENCE DIAGRAM

In order to have a better understanding of the system, I would provide a few series of sequence diagrams here to illustrate the main functions. According to Figure 2, the main function of the system can be concluded for user as registration, verification, and identification.

### 4.2.1 Registration

Figure 4 shows the sequence diagram of registration function. A user should first register his own information so that he can log in the next time. Therefore, the user would send the request of registration to the Interface Module. Then the Interface Module will receive the request and ask for the user to provide his/her palm vein to the Sensor Module and Sensor Module would send captured image back to Interface Module. After the Interface Module receives the image, it will transfer the user's information to the database. Finally, a status message would be transmitted back and shown by Interface Module.

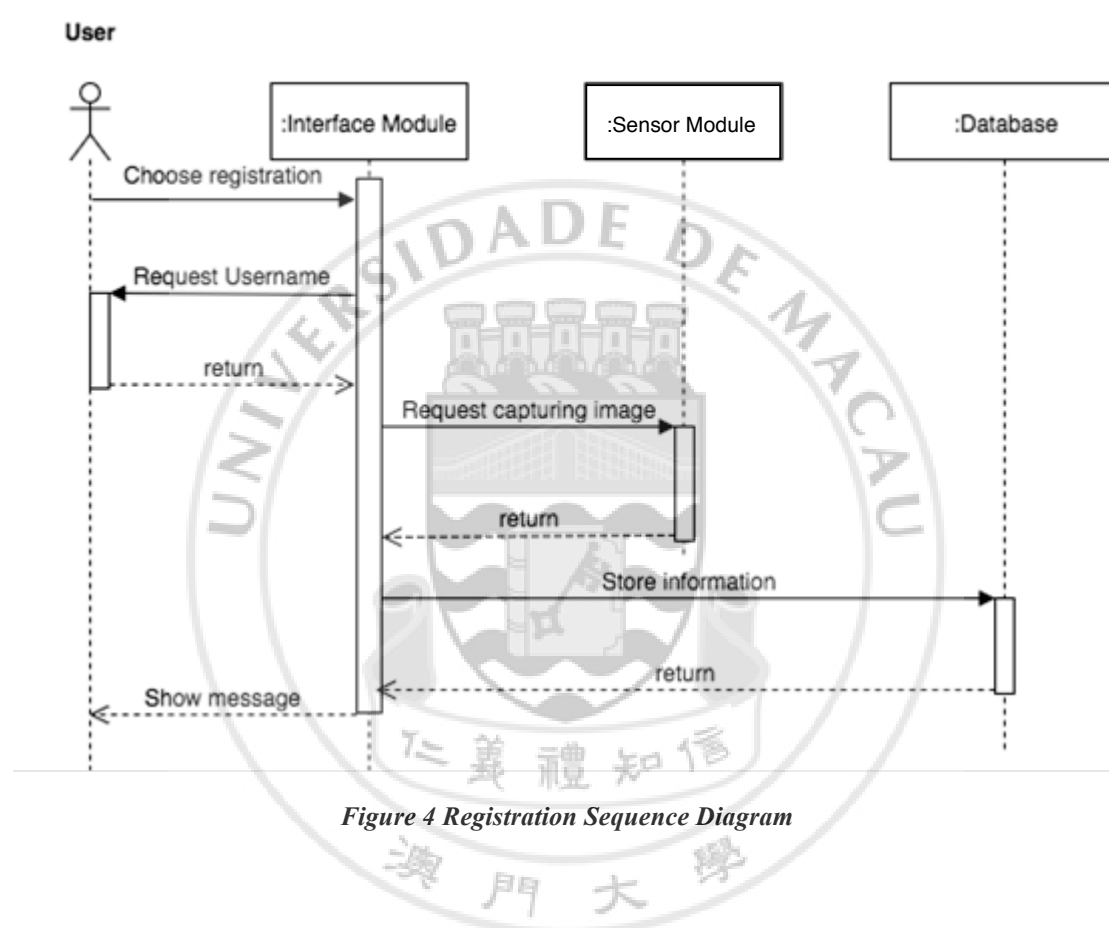
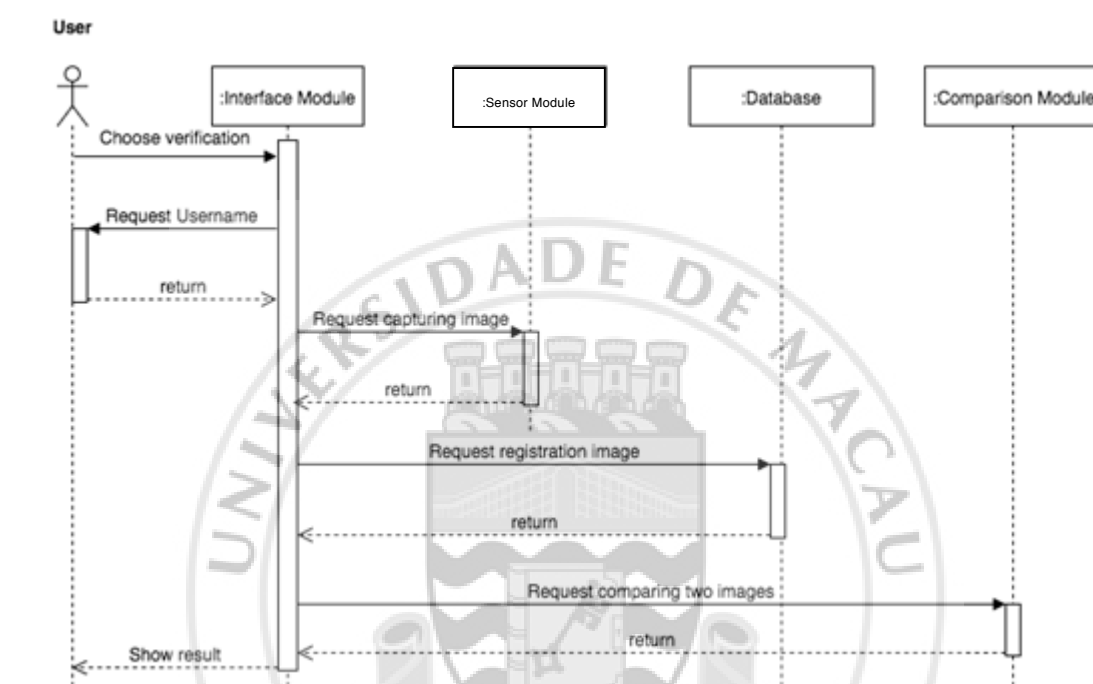


Figure 4 Registration Sequence Diagram

### 4.2.2 Verification

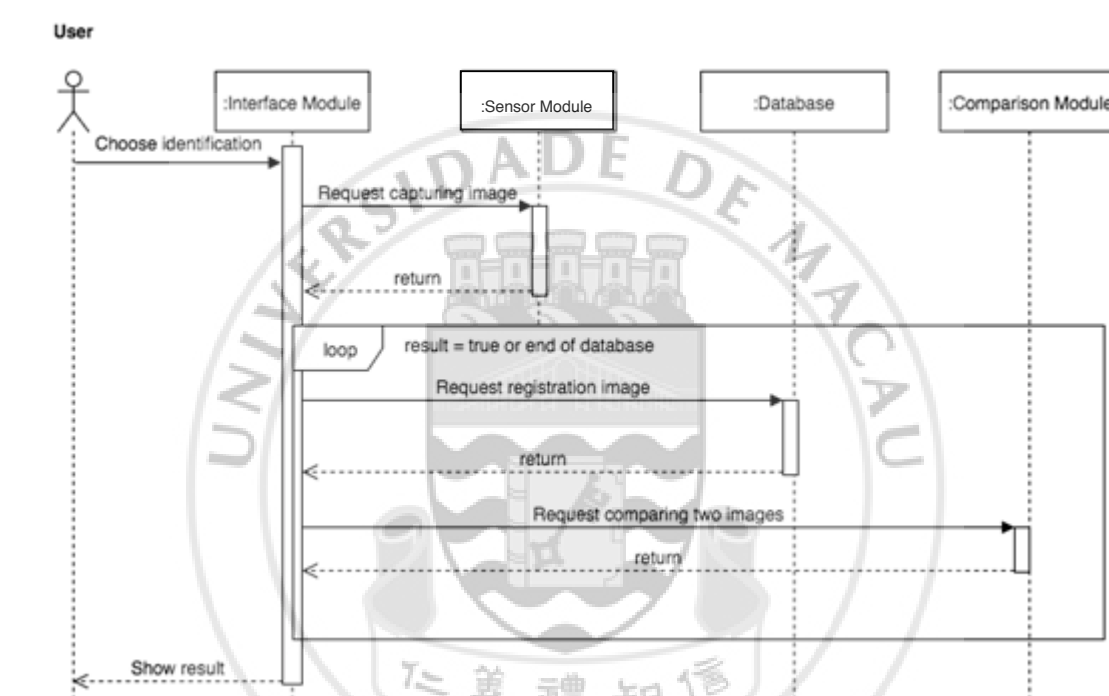
Figure 5 represents the sequence diagram of verification function. Four modules are involved in this function. First the user should choose the verification mode to log in. Then the Interface Module would request the user to fill in a username. Then Sensor Module would be called to capture the user's palm vein. In addition, the registered image would be retrieved from the database. At last, the two images would be sent to Comparison Module to check whether they are a match and the result would be sent back to Interface Module and shown to the user.



*Figure 5 Verification Sequence Diagram*

### 4.2.3 Identification

Figure 6 is the sequence diagram of identification function. The main part of it is similar to verification. The user should first choose to log in by identification. Then Sensor Module would request the user to directly provide his/her palm vein image without typing in username. Then the system would encounter a loop on the basis of number of images in the database until the output result is a match or it reaches the end of the database. In the loop, Interface Module would first retrieve a registration image and sent it together with the user's palm vein image to Comparison Module to check out whether they compose a match. If they do match, the loop will end and output the positive result. Or else, the loop will continue or come to an end and output failure when there are no more images in the database.



*Figure 6 Identification Sequence Diagram*

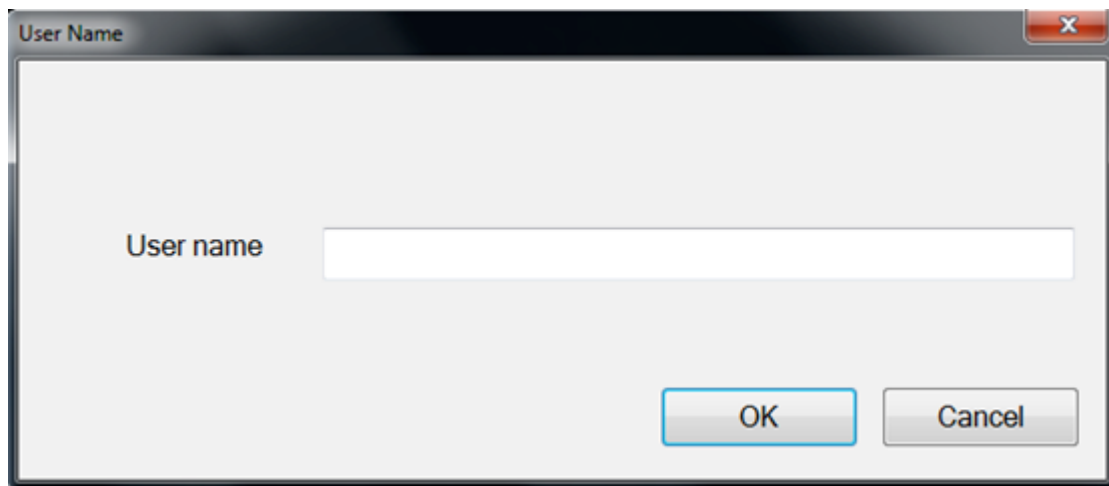
### 4.3 Interface description

The basic procedure of my application is divided into two parts. The first part is function of registration and the second part is function of recognition. This is the main interface of my program. There are four buttons on the interface. If you press 'Capture', the camera would be turned on and start capturing images. Press 'Help' and you will get some detailed information of my project. 'Register' and 'Log in' correspond to the two procedures of my program respectively.



*Figure 7 Main Interface*

First is the registration stage. If you press 'Register' button, a new dialog will pop up requiring your user name.



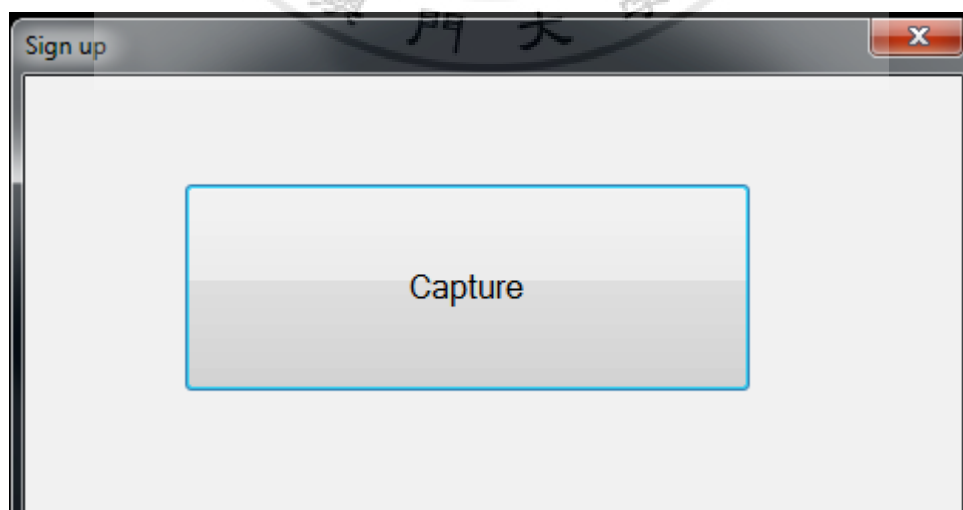
*Figure 8 Register Interface*

If you do not type anything, a warning message dialog will pop up to inform you of the necessity of typing your user name.



*Figure 9 User Name Warning*

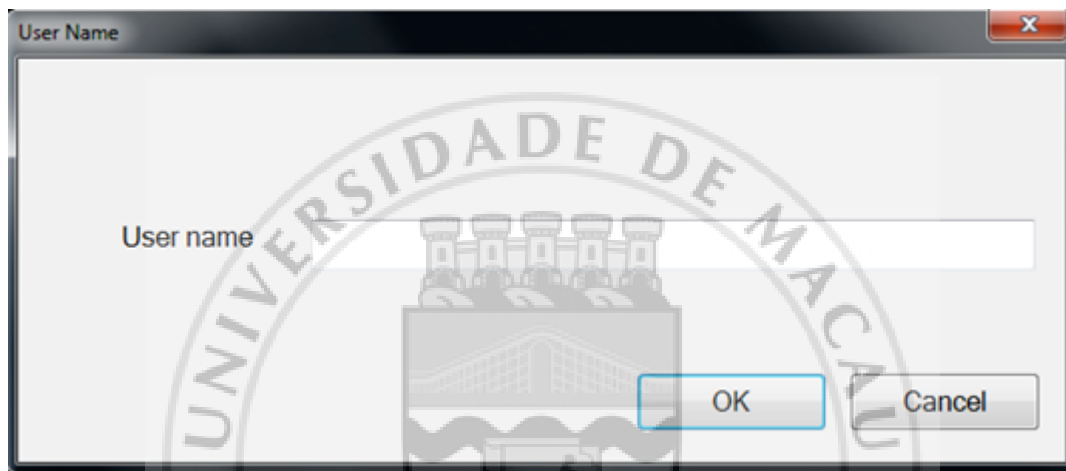
After you put in your user name, click OK. A small dialogue window will appear with just one button 'Capture' on it. Now please put your hand on our device and fit your finger well into it. Then click Capture, your personal information and hand vein picture would be extracted to our database here.



*Figure 10 Capture Button*

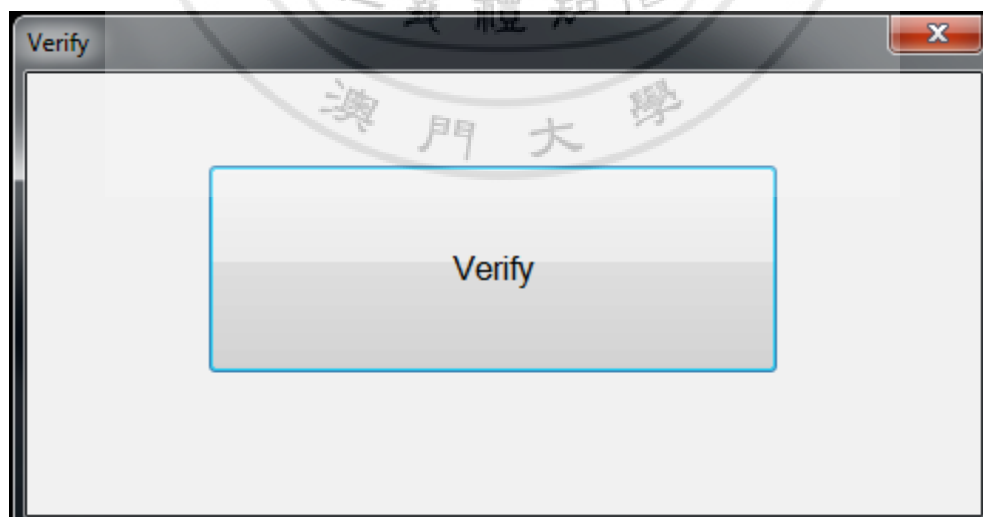
When we proceed to the recognition stage, I have developed two modes for recognizing. The first one is verification mode, which means the user would type in his name first and then the system would test whether the person verified is genuine or imposter. The second one is identification mode, which means one to many matches. The user does not need to put in his name and he can just put his hand on the device and wait for the system to recognize whether the user is a registered one in our database or a non-registered one. Generally, the identification mode is more convenient but complicated because it requires a large number of comparisons. With the database growing, the identification mode will cost much longer time.

I will first introduce the verification mode. First click Log in. This mode requires the user to type in his username as he did in registration step.



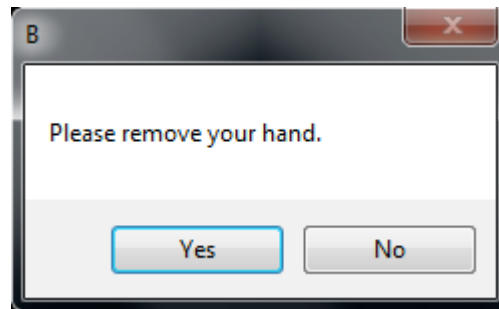
*Figure 11 Verification Interface*

Click OK and it will proceed to capturing user's current palm vein. The user can put his hand on our device again and click Verify.



*Figure 12 Verify Button*

If the program has successfully captured the user's palm print, a message box will pop up indicating that the user can move his hand now.



*Figure 13 Capture Complete Notification*

After capturing the palm vein of the user, the program would automatically figure out whether it is the matched image with the registered one in the database. If the matching phase is successfully completed, the whole screen would turn to green and user's welcome interface would appear like Figure 14.



*Figure 14 Verification Success Notification*

Or else, a red screen would appear instead and warn you that verification has failed like Figure 15.





*Figure 15 Verification Failure Notification*

Identification could be a much faster way to recognize your information with just one more step of typing your username. For example, customs between Macau and Zhuhai also applies this method to automatically check people's identification by checking their passports first and then extract their palm prints. However, if it were possible to bypass this procedure, this device would automatically search in the database whether there is a matching palm vein with the current user. I also implemented this function in the program and it can successfully detect the user's identification if he has registered before.

Click Log in (Identification) and it will pop up directly a dialogue requiring your palm vein. User can put his hand on the device and click Identification.



*Figure 16 Identify Button*

If the program has successfully captured the user's palm print, a message box will pop up indicating that the user can move his hand now. Then, the identification result will show on the screen just as verification mode mentioned before.



## CHAPTER 5. Implementation Narratives and Description

In this chapter, a more detailed description of algorithm and implementation of the system will be discussed.

### 5.1 IMAGE PROCESSING

This is the most important part of our system. The precision of recognition would depend greatly on this part. I would introduce the most important steps applied in my algorithm.

#### 5.1.1 ROI (Region of Interest)

Since the camera would capture the whole shape of the user's hand. What we are interested in is the fixed palm area with veins under the skin. Therefore, we chose the central 320\*320 pixels as our region of interest shown in Figure 17.



Figure 17 ROI of Original

#### 5.1.2 Gaussian Filter

Gaussian filter is a filter whose impulse response is a Gaussian function (or an approximation to it). Gaussian filters have the properties of having no overshoot to a step function input while minimizing the rise and fall time. This behaviour is closely connected to the fact that the Gaussian filter has the minimum possible group delay. It is considered the ideal time domain filter. The 0 mean 1-D Gaussian filter kernel has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

After deciding proper  $\sigma$ , we still need another parameter  $d$ .  $d$  is the number of directions we use to apply to our captured image. As the vein distribution in our palm towards a lot of directions, it is fair that we use several 1-D Gaussian filter to process

the image. Figure 18 is the Gaussian function with various  $\sigma$ . With  $\sigma$  increasing, we can see the shape of the Gaussian function becomes smoother.

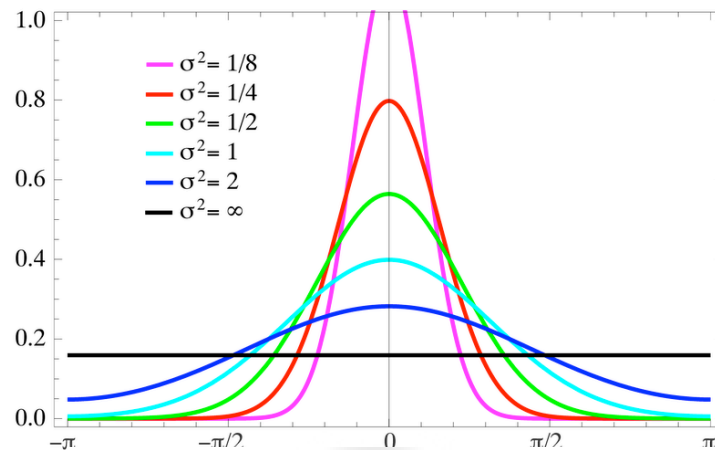


Figure 18 Gaussian Function

After we retrieve proper Gaussian filter, we then use it to interact with our image, the process of which is called convolution. Convolution is not traditional matrix multiplication, despite being similarly denoted by  $*$ . It is frequently used in the process of adding each element of the image to its local neighbours, weighted by the kernel. In our case, we use Gaussian kernel to do the convolution. However, the Gaussian function we use should reverse its vertical direction since we would like the vein part in the final result to be white pixels and others to be black pixels.

The reason we choose Gaussian filter to deal with the palm vein image is that the shape of 1-D Gaussian function is very similar to shape of palm vein. It is necessary to pick a proper  $\sigma$  and number direction. First I would like to choose a proper  $\sigma$ . Figure 19 shows that when  $\sigma$  equals 4, most details can be well captured and when  $\sigma$  equals 10, the main vein can be maintained while not losing too many details. Due to the lack of quality of camera, we need to prune the result as precise as possible. Therefore, we choose  $\sigma$  to be a combination of 4 and 10.

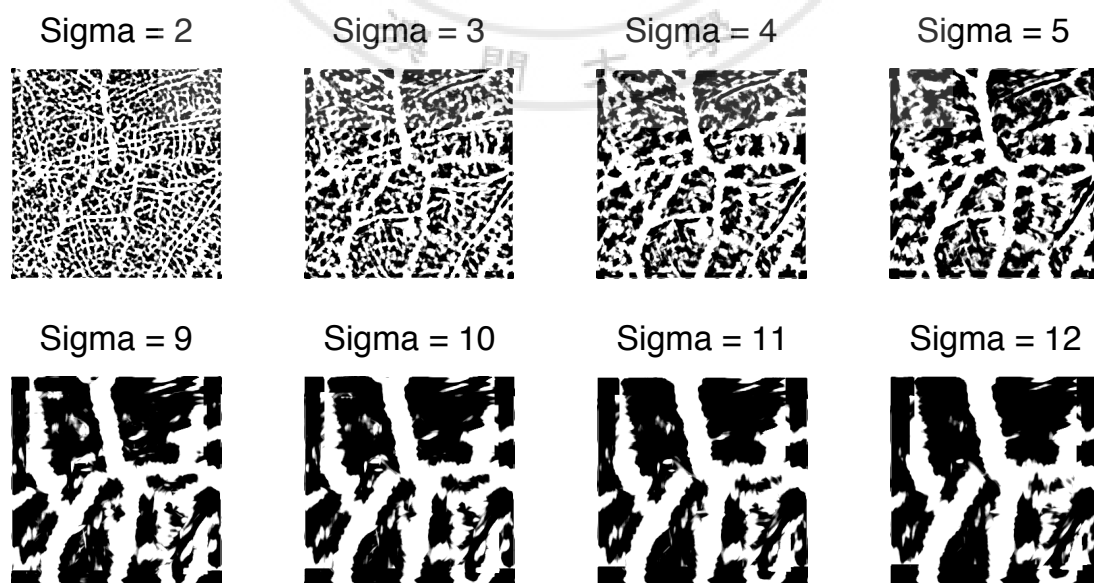
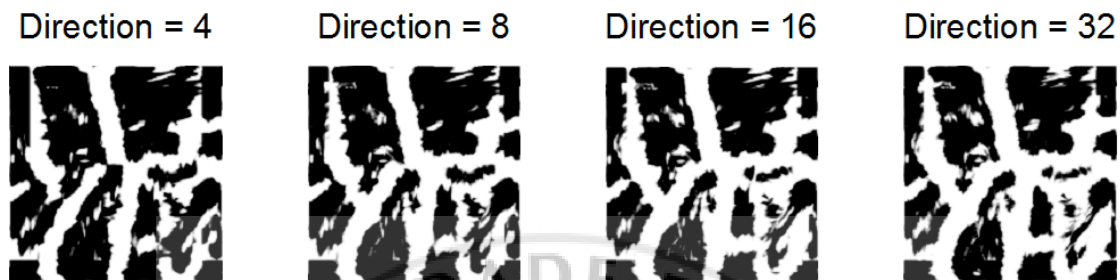


Figure 19 Gaussian Filter Result with Different  $\sigma$

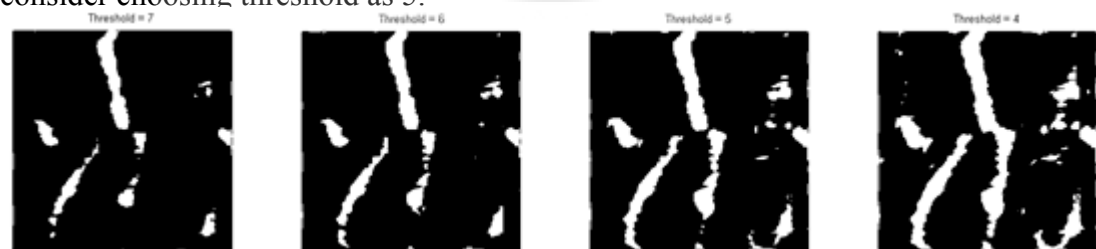
Then, we need to pick a proper number of direction  $n$ . Let us focus on the case that  $\sigma$  equals 10. From Figure 20, we can see that there is no big difference with the increasing of direction. In order to reduce the computational cost, we would choose direction to be 8. After detecting palm veins through 8 directions, eight images would be generated and combined into one final result by choosing the highest value in the eight images.



*Figure 20 Gaussian Filter Result with Different Directions*

### 5.1.3 Transform Grey Level to Binary

It is essential to convert an image from its grey level to binary in order to compare the similarity of this image to other images. Since grey level image has a wide range of value, it is not convenient to compare one grey level image with another one. By means of comparing binary images, we can easily compute the similarity of two images simply by summing all the pixels that have the same value, which can either be 0 or 1. There are many methods available to generate a binary image from a grey image but in our case we can simply set a threshold to realize this function, if one pixel value is larger than the threshold, then it would be transformed to 1. On the other hand, if the pixel value were smaller than the threshold, it would turn to 0. This simple method would definitely generate a lot of noisy points, where a lot of small-scale white area would be also transferred to white as our target value. Figure 21 shows binary images under different thresholds. We can find that if we choose threshold smaller than 5, then too many noisy regions exist in the image. If we choose threshold bigger than 5, then the vein in the image starts to split. Therefore, we consider choosing threshold as 5.



*Figure 21 Binary Transformation with Different Thresholds*

As a result, we figure out the threshold algorithm in Equation (2)

$$\text{Binary Image} = \begin{cases} 1 & \text{pixel value} \geq 5 \\ 0 & \text{pixel value} < 5 \end{cases} \quad (2)$$

#### 5.1.4 Eliminating Small Image Regions

In order to eliminate the effect of small-scale white area that does not actually belong to palm vein, we need to remove those small regions. The algorithm is not so difficult and we can use a series built-in function of MATLAB to accomplish this task. First, I would use *bwlabel* to find all the connected area in the image and make a label for each of them. There are two alternatives of ways to identify connectivity, 4 directions or 8 directions. Referring to Figure 22, it is better to choose 8 directions since the area would stretch in various directions.



Figure 22 Binary Palm Vein with Many Noisy Regions

Then, we can use function *regionprops* to eliminate regions that contain too few pixels in it. Since we have two separate images, we need to set different thresholds for them individually. We can easily find from Figure that the image with smaller  $\sigma$  has much smaller noisy area and the vein region is also smaller. So we should set a smaller threshold for image with smaller  $\sigma$  and vice versa. After throwing noisy regions, we could see from Figure 23 that the images are much clearer than before.



*Figure 23 Binary Palm Vein after Eliminating Small Regions*

However, we could find there are noisy vertical lines on the leftmost and rightmost part of the images caused by low quality of camera. We will crop them out in the later section.

### 5.1.5 Image Registration

Figure shows the two palm vein images from the same person. However, it is easy to see from Figure 24 that position transition exists between them and if we directly use these two images to conduct comparison, the result would not be so reasonable. As a result, registration is needed to move one picture to a reasonable position and then proceed to comparison.



*Figure 24 Binary Palm Vein before Registration*

We used the built-in function of MATLAB to conduct image registration and here is the main code of it.

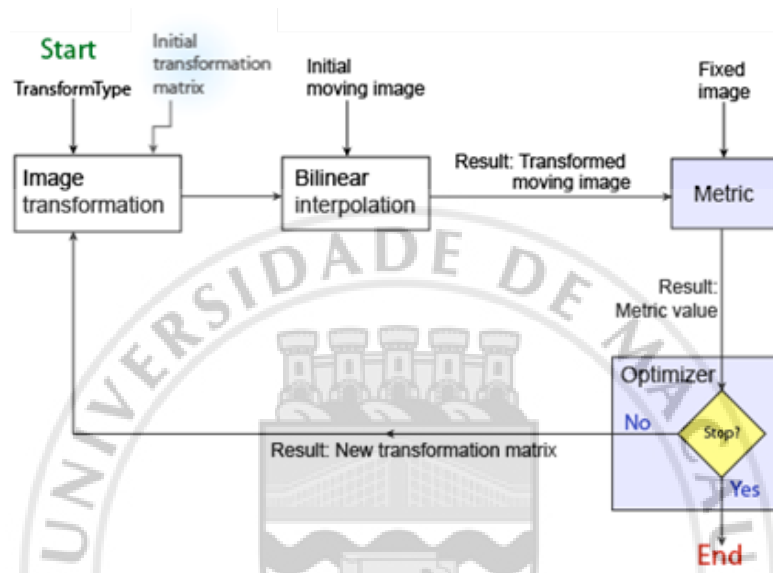
```
[~, metric] = imregconfig('monomodal');
optimizer = registration.optimizer.OnePlusOneEvolutionary();
optimizer.InitialRadius = optimizer.InitialRadius/3.5;
```

```

movingRegisteredDefault = imregister(result2, result1, 'translation',
optimizer, metric);

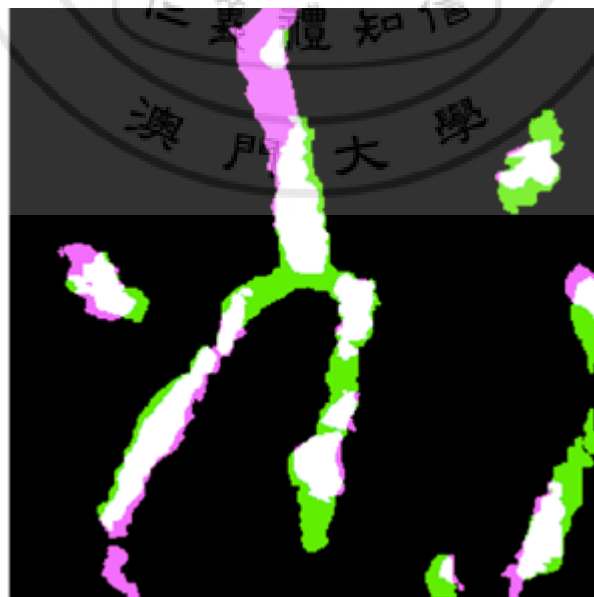
```

The first line returns the measurement of registration *metric*. The second and third line generates an optimizer for the measurement and set the initial step smaller to rise up the precision. Figure 25 shows the main procedure of intensity-based registration [9]. It begins with select a transformation type and then the similarity metric value is computed. At last, the optimizer will check for a stop condition. If it does not satisfy the condition, a new transformation matrix would be generated and loop will continue.



**Figure 25 Registration Flow Chart**

Finally, the last line performs registration and image *result1* will be registered to *result2* by means of translation and measurement provided before. The result turns out well registered as shown in Figure 26.



**Figure 26 Binary Palm Vein Combination after Registration**



Green is the fixed palm vein and pink is the moved palm vein. White is the superposition of two images.

## 5.2 Comparison

After all these image processing steps, we need to calculate the similarity of two palm vein images. The simplest way is to count how many pixels are the same between two images and divide it by the total number of pixels. However, after experimenting, we found that the value of similarity is always very high because the background is black and the most number of pixels are the same.

In order to solve this problem, we found Jaccard index [10] suit our case well. Given two images A and B, the attributes of which are either 0 or 1. Let  $M_{11}$  represents the total number of attributes where A and B both have a value of 1.  $M_{00}$  represents the total number of attributes where A and B both have a value of 0.  $M_{01}$  represents the total number of attributes where the attribute of A is 0 and the attribute of B is 1.  $M_{10}$  represents the total number of attributes where the attribute of A is 1 and the attribute of B is 0.

$$Jaccard = \frac{M_{11}}{M_{01} + M_{11} + M_{10}} \quad (3)$$

Equation (3) defines Jaccard similarity. This is more proper for our system since the denominator exclude the back ground part. Therefore, the Jaccard similarity will vary obviously and give us a clear sense of the separation plane.

However, due to the instability of Sensor Module and illumination factor, it is dangerous to depend on only one variable. Therefore, in our case, I separate Jaccard similarity into two part.

$$Similarity1 = \frac{M_{11}}{M_{10} + M_{11}} \quad (4)$$

$$Similarity2 = \frac{M_{11}}{M_{01} + M_{11}} \quad (5)$$

### 5.3 SVM Classifier

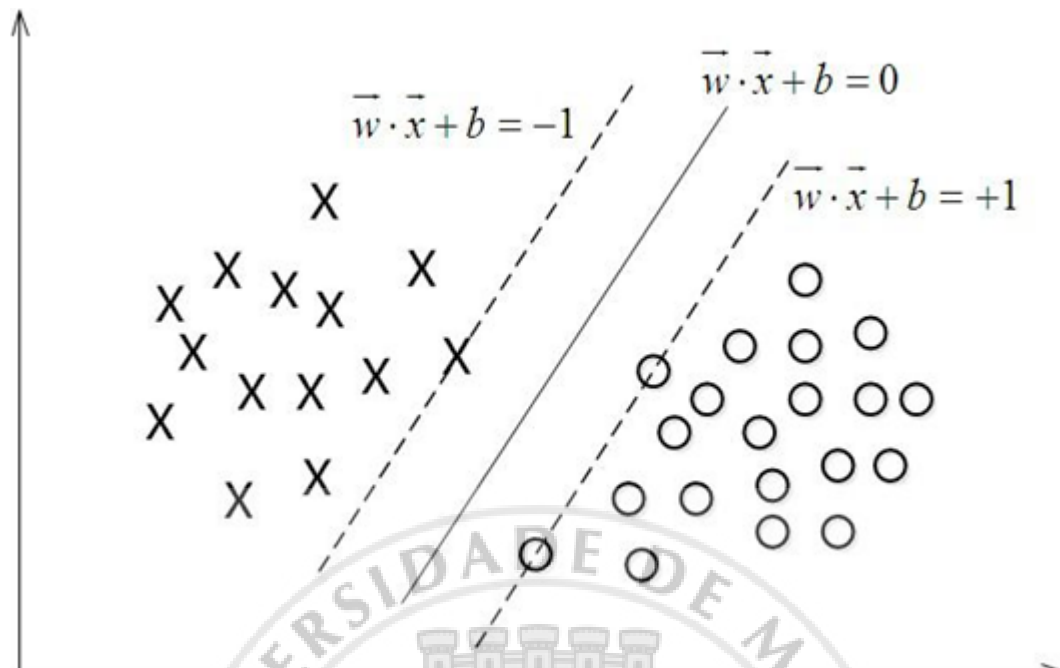


Figure 27 SVM Classifier

As we can see from Figure,  $\vec{w} \cdot \vec{x} + b = 0$  is the hyperplane separating points of two classes. The width of the hyperplane can be derived by

$$\text{Width} = (x_+ - x_-) \cdot \frac{w}{\|w\|} \quad (6)$$

Therefore, the objective function is to maximize  $\frac{2}{\|w\|}$ .

During the process of solving the extreme out, the optimization function is as follows

$$L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \quad (7)$$

Also, the decision rule can be derived by

$$\begin{cases} \text{Positive} & \sum_i \alpha_i y_i x_i x + b \geq 0 \\ \text{Negative} & \sum_i \alpha_i y_i x_i x + b < 0 \end{cases} \quad (8)$$

Equation and Equation are all decided by on the dot product of paris of examples  $x_i x_j$ . So if we would like to retrieve a classifier to classify not linearly separable data, we can transfer the dot product of examples to a higer dimension by means of choosing a proper kernel function.

Our project has successfully done experiments on 34 volunteers. Therefore, the total number of data points should be  $34 \times 34 = 1156$ , which is a quite large number. It is

hard to doing classification with our eyes so we make good use of SVM classifier to help us train the data and generate a classifier for future classification task.

SVM is a supervised learning model used for classification or regression analysis. It trains a set of labeled data and tries to find the maximum-margin hyper plane that can separate these data well.

In our case, we applied the built-in function of MATLAB *svmtrain* to develop SVM model and use it to make prediction. Since we have altogether 1156 sets of data, we would like to choose 50% of the data to be training data and 50% of the data to be test data. Figure shows the training result of SVM with Quadratic kernel, RBF kernel and linear kernel.



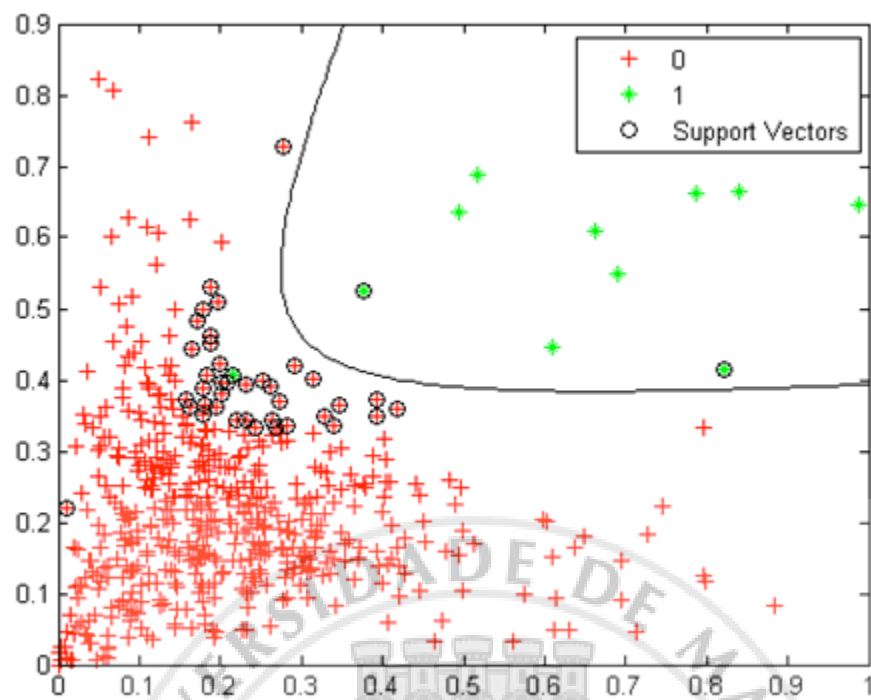


Figure 28 Quadratic Kernel

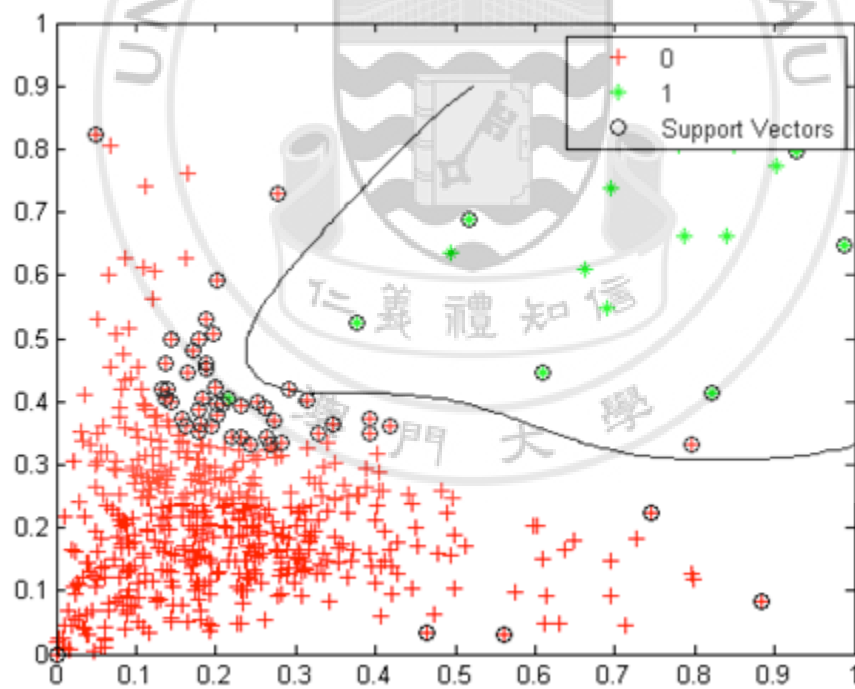
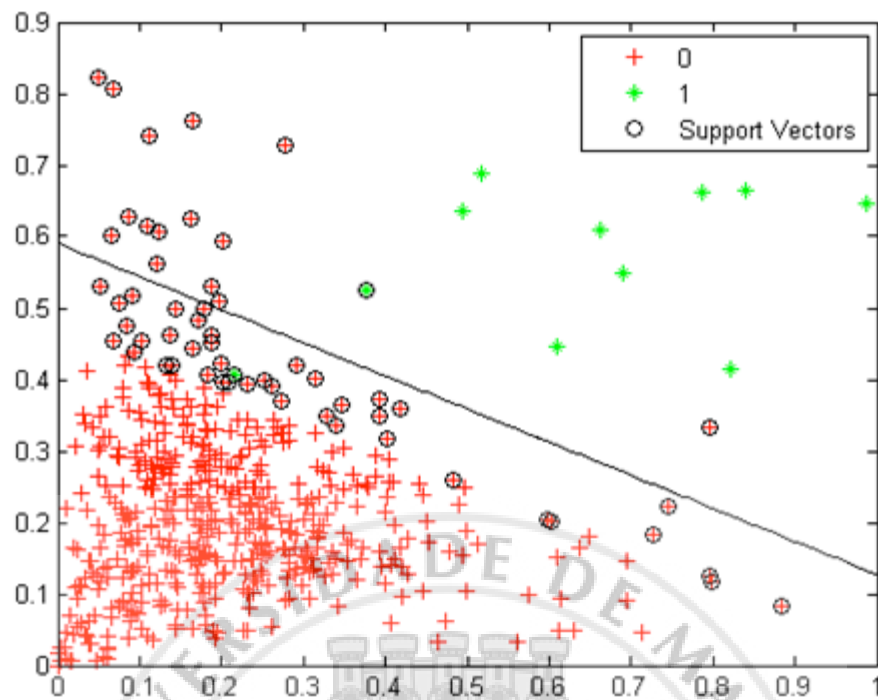


Figure 29 RBF Kernel



*Figure 30 Linear Kernel*

After training the model, we would then test the performance of the models by applying the models on the test set.

*Table 2 SVM Kernel Comparison*

Kernel Name	Precision
Linear	97.2318%
Quadratic	99.1349%
RBF	99.3080%

As a result, RBF kernel has a better performance than linear kernel and we would RBF as our kernel of the final model.

## 5.4 MFC Framework

The Microsoft Foundation Class Library (MFC) is an "application framework" for programming in Microsoft Windows. MFC provides most of the essential code, like managing windows, menus and dialog boxes, various data types. It makes it easier to create an application and put all the necessary stuff into a box. There is little chance for us to program for windows, all we need are basic C++ skills and understanding framework of MFC. As a result, it greatly shorten the development time and makes codes more portable. Our application makes good use of MFC framework and creates the interface of the recognition system.

## 5.5 Multiple Platform

In order to make good use of MATLAB's usability in handling matrix and a lot of built in functions, we need to construct a multi-platform in this case.

First we need to use the compiling tool *mex* on MATLAB to compile the .m file into .lib, .dll, .h which are necessary to put the program into use in C++. Then we should set up the environment to call the compiled file and use *mwArray* as the communicative data structure between MATLAB and C++.

However, it will cost much more time to call MATLAB program from C++, so as we will mention later, one possible future improvement may transform MATLAB directly to C++ code, which would increase the speed of the program enormously.



## CHAPTER 6. System Quality

In this chapter, various tests would be carried out to do the verification test and ensure the quality of the system. There are altogether four test cases, 1) a genuine individual is accepted in verification mode, 2) an imposter individual is rejected in verification mode, 3) a genuine individual is accepted in identification mode, 4) an imposter individual is rejected in identification mode.

### 6.1 A Genuine Individual in Verification Mode

*Table 3 Test Case 1*

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	<p>Verification Test, To verify that the verify function performs correctly after the ‘verify’ button is pushed, the result of recognition would be shown in the form of color and text message.</p> <p>Defect Test, To verify that the system could handle several abnormal situations as, a warning message should be displayed on the screen if the user does not provide his/her user name or provides the name that does not exists in the database.</p>
Prerequisites	The system under normal state.
Test Procedure	<p>Verification Test,</p> <ol style="list-style-type: none"> <li>1. The user will provide his/her user name and his/her palm vein image to the system.</li> <li>2. Click on the “verify” button and wait for final result.</li> </ol> <p>Defect Test,</p> <ol style="list-style-type: none"> <li>1. The user will leave the user name blank or types in the non-existent name and wait for the response of the system.</li> </ol>
Test Data	User’s user name and palm vein image.
Result	<p>Verification Test,</p> <ol style="list-style-type: none"> <li>1. A screen of green and the user name will appear.</li> </ol>

	Defect Test,  1. A warning message would be displayed saying, “Please fill in your user name” or “Sorry you haven’t signed up”.
Date of Creation	7/5/2017

## 6.2 An imposter in Verification Mode

*Table 4 Test Case 2*

Test Suite ID	TS002
Test Case ID	TC002
Test Case Summary	<p>Verification Test, To verify that the verify function performs correctly after the ‘verify’ button is pushed, the result of recognition would be shown in the form of color and text message.</p> <p>Defect Test, To verify that the system could handle several abnormal situations as, a warning message should be displayed on the screen if the user does not provide his/her user name or provides the name that does not exists in the database.</p>
Prerequisites	The system under normal state.
Test Procedure	<p>Verification Test,</p> <ol style="list-style-type: none"> <li>1. The user will provide a fake user name and his/her palm vein image to the system.</li> <li>2. Click on the “verify” button and wait for final result.</li> </ol> <p>Defect Test,</p> <ol style="list-style-type: none"> <li>1. The user will leave the user name blank or types in the non-existent name and wait for the response of the system.</li> </ol>
Test Data	User’s user name and palm vein image.
Result	<p>Verification Test,</p> <ol style="list-style-type: none"> <li>1. A screen of red and the text message would be displayed saying, “Verification failed, please try again”.</li> </ol>



	Defect Test, 1. A warning message would be displayed saying, “Please fill in your user name” or “Sorry you haven’t signed up”.
Date of Creation	7/5/2017

### 6.3 A Genuine Individual in Identification Mode

*Table 5 Test Case 3*

Test Suite ID	TS003
Test Case ID	TC003
Test Case Summary	Verification Test, To verify that the identification function performs correctly after the ‘Identify’ button is pushed, the result of recognition would be shown in the form of color and text message.
Prerequisites	The system under normal state.
Test Procedure	Verification Test, 1. The user will provide his/her palm vein image to the system. 2. Click on the “Identify” button and wait for final result.
Test Data	User’s palm vein image.
Result	Verification Test, 1. A screen of green and the user name will appear.
Date of Creation	7/5/2017

### 6.4 An Imposter in Identification Mode

**Table 6 Test Case 4**

Test Suite ID	TS004
Test Case ID	TC004
Test Case Summary	Verification Test, To verify that the identification function performs correctly after the 'Identify' button is pushed, the result of recognition would be shown in the form of color and text message.
Prerequisites	The system under normal state.
Test Procedure	Verification Test, <ol style="list-style-type: none"> <li>1. The user will provide his/her palm vein image, which does not exist in the database to the system.</li> <li>2. Click on the "Identify" button and wait for final result.</li> </ol>
Test Data	User's palm vein image.
Result	Verification Test, <ol style="list-style-type: none"> <li>1. A screen of red and the text message would be displayed saying, "Identification failed, please try again".</li> </ol>
Date of Creation	7/5/2017

## CHAPTER 7. Ethics and Professional Conduct

### 7.1 Use of Data

The data of this project consists of two parts, user name and palm vein image. It is one kind of biological information, which can represent the identity of an individual person. If the palm vein information was stolen by someone and used for illegal behaviour, it might lead to severe result. Therefore, keeping the data preserved well is of great importance in our project.

At the time we collected the subjects' information, we required their student number instead of English name for fear of any duplication of name in the database. However, when the project is presented or exhibited in the future, we will try to protect these data containing personal information. For example, the user names would be transformed into a fake name like Jeremy → Subject1.

The experimental data would be kept well later and encrypted to avoid being used for any other purpose.

### 7.2 Coding

In this part, I will give extra credit for those who provide me with necessary code and help me finish this program.

- *Bob Zhang – MatchFilterWithGaussDerivative.m*

This program helps generate Gaussian filter given the value of  $\sigma$  and number of directions.

- *MATLAB built-in function – conv2, imregister, regionprops, svmtrain*

MATLAB is of great use when it comes to the process of images. Various built-in functions have saved a lot of time for me to accomplish this project.

### 7.3 Miscellaneous

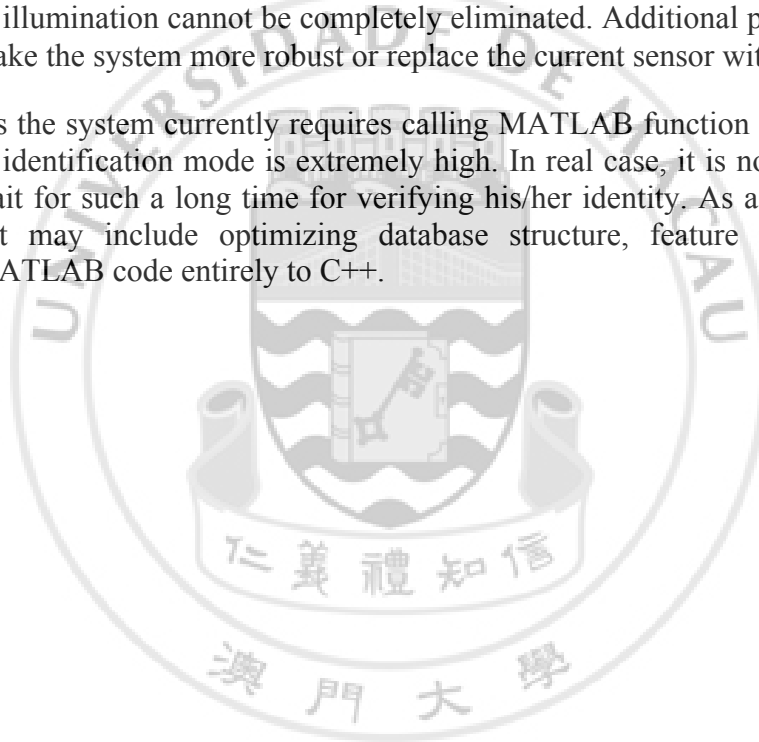
All the data is collected within University of Macau and will be only used for research objective. Thus, any commercial goal won't be allowed throughout this project. In addition, all the content mentioned in the design and implementation are accomplished by me with my supervisor Bob Zhang, which are authentic and accurate. No fake data and statement of content would appear in the report.

## CHAPTER 8. Summary

The system presented in this report is a palm vein recognition system constructed with C++ and MATLAB. The features that make our system novel are: i) very low cost, ii) excellent performance in the precision of recognition. Gaussian filter, binary image transformation, eliminating small regions and registration help to process the image to generate the similarity value of two images in a better way. In addition, SVM classifier with a proper kernel helps to classify the similarity so that a final result would be calculated out.

The system was tested using the real data of 34 subjects collected by us. The accuracy rate 99.3080% shows the high performance of this system. The failure cases may be caused by severe shifting of the subject or various illumination situation. Although we have placed a black box to protect the Sensor Module from too many light coming in, the factor of illumination cannot be completely eliminated. Additional process may be needed to make the system more robust or replace the current sensor with a better one.

Moreover, as the system currently requires calling MATLAB function from C++, the time cost of identification mode is extremely high. In real case, it is not practical for people to wait for such a long time for verifying his/her identity. As a result, further improvement may include optimizing database structure, feature matching and transform MATLAB code entirely to C++.



## CHAPTER 9. REFERENCES

- [1] Kono M, Ueki H, Umemura S. Near-infrared finger vein patterns for personal identification[J]. *Applied Optics*, 2002, 41(35): 7429-7436.
- [2] Wang Z, Zhang B, Chen W, et al. A performance evaluation of shape and texture based methods for vein recognition[C]//Image and Signal Processing, 2008. CISP'08. Congress on. IEEE, 2008, 2: 659-661.
- [3] Mirmohamadsadeghi L, Drygajlo A. Palm vein recognition with local binary patterns and local derivative patterns[C]//Biometrics (IJCBI), 2011 International Joint Conference on. IEEE, 2011: 1-6.
- [4] Yang G, Xi X, Yin Y. Finger vein recognition based on a personalized best bit map[J]. *Sensors*, 2012, 12(2): 1738-1757.
- [5] Rosdi B A, Shing C W, Suandi S A. Finger vein recognition using local line binary pattern[J]. *Sensors*, 2011, 11(12): 11357-11371.
- [6] Peng M, Wang C, Chen T, et al. A methodology for palm vein image enhancement and visualization[C]//Online Analysis and Computing Science (ICOACS), IEEE International Conference of. IEEE, 2016: 57-60.
- [7] Gupta P, Srivastava S, Gupta P. An accurate infrared hand geometry and vein pattern based authentication system[J]. *Knowledge-Based Systems*, 2016, 103: 143-155.
- [8] Cao J, Xu M, Shi W, et al. MyPalmVein: A palm vein-based low-cost mobile identification system for wide age range[C]//E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on. IEEE, 2015: 292-297.
- [9] Khunt D P, Makwana Y N. Image registration using intensity based technique[J]. *Journal of information, knowledge and research in electronics and communication engineering*, 2012, 2(2): 607-609.
- [10] Kane N, Aznag K, El Oirrak A, et al. Binary data comparison using similarity indices and principal components analysis[J]. *Int. Arab J. Inf. Technol.*, 2016, 13(2): 232-237.