

**Functional Specification**

**<Fingerprint Recognition>**

Supervisor: Nigel Whyte

Student: Dayu Chen

Student ID: C00131022

Submit date: 2010-12-16

Contents

[1. Introduction 1](#_Toc280198050)

[2. Application Prototype 1](#_Toc280198051)

[3. Functional Description 3](#_Toc280198052)

[3.1 Core functional 3](#_Toc280198056)

[3.1.1 Load fingerprint image 3](#_Toc280198057)

[3.1.2 Image pre-processing 3](#_Toc280198058)

[3.1.3 Recognition. 4](#_Toc280198059)

[3.2 Additional functional 5](#_Toc280198060)

[4 Potential users 5](#_Toc280198061)

[5 Metrics 5](#_Toc280198062)

# Introduction

This document has been written to explain what functions my fingerprint recognition application will have and how they work. The functional is categorized into two: core functional and additional functional. The core functional in my application is recognition. In order to let user to understand the processing of recognition, the core functional should be divided into three functions: load fingerprint image, Image pre-processing and image recognition. Additional functional may not be the key functional in this program, but still important. The additional functional is fingerprint image management. This functional is a database management functional, it allow user to manage a fingerprint image in system database (e.g. enroll, delete and update).

# Application Prototype

This part shows the prototype of application GUI, in order to explain the functional more clearly. Figure 1 is the recognition user interface. Core functional is included in this window.

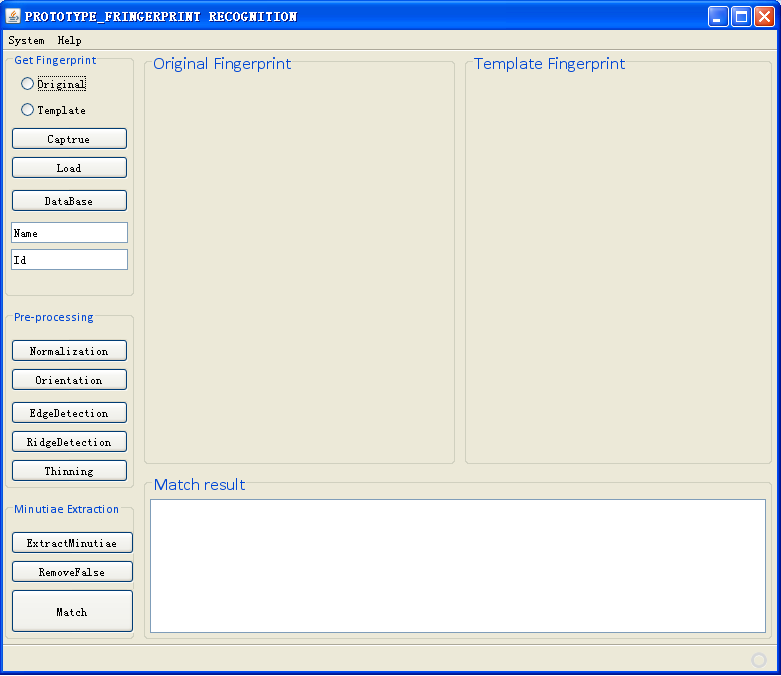


Figure 1 Fingerprint recognition system: user interface of recognition

In Left of Figure 1, there are three panels: Get Fingerprint, Pre-processing and Minutiae Extraction. These three panels are for the three cores functional: load fingerprint image, Image pre-processing and image recognition. Each core function has divided to many sub-functions. In loading fingerprint image, user has to load two fingerprint images, one is for original fingerprint and one is for template fingerprint. After user load two fingerprint images, these two images will display in the original fingerprint panel and template fingerprint panel, and then user can execute the pre-processing function. In pre-processing, all the process has been divided into five steps, each button in pre-processing panel instead each step in image pre-processing. User only needs to press these buttons one time, images in original fingerprint panel and template fingerprint panel will do the step at the same time. The result of each step will be displayed in original fingerprint panel and template fingerprint panel. The “ExtractMinutiae” button and “RemoveFalse” button works the same. Match is the last step of recognition, the match result will be displayed in the Match result panel, and it shows the final result to tell user whether original fingerprint and the template fingerprint is the same fingerprint. It not only shows the final result, but also shows the same minutiae, including local position, direction and minutia type.

At the top of this window, there is a menu called system, user can go to the additional function with this menu. Figure 2 is the window of enroll a fingerprint.

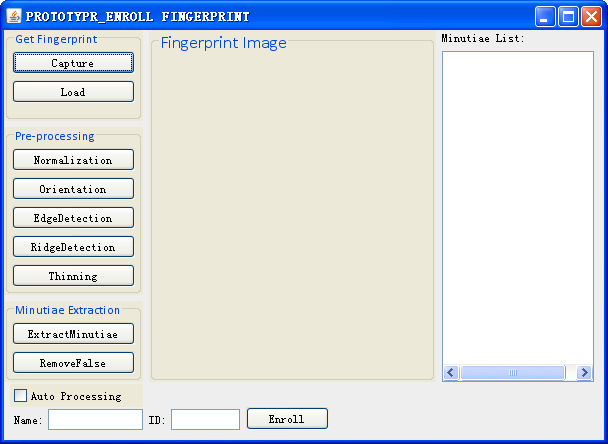


Figure 2 Fingerprint recognition system: User interface of Enroll fingerprint

Fingerprint management is the additional functional of my program. This functional include three sub-functions: fingerprint enrolls, update and delete. Fingerprint enrolls is important to another two functions. In Figure 2, the main processing is like core functional, user can load image, do pre-processing and minutiae extraction. Image and result of each step will be showed in the “Fingerprint Image” panel. After minutiae extraction, all minutiae of the fingerprint will be showed in the result list. In this list, every minutia is in the structure like that: minutia ID, local position, direction and type. User only can enroll fingerprint after minutiae extraction. User also should write down some information for the fingerprint (e.g. name and ID). In enroll; user can let system to do the pre-processing automatically. In fingerprint update, user only can update the extra information for a fingerprint (e.g. name and ID). User does not allow updating minutiae for fingerprint.

# Functional Description

This part is to make description to each functional, in order to explain them clearly.



## Core functional

### Load fingerprint image

This function is for import fingerprint image for processing. This is the basic function in this application. In this function, system provides three approaches to import the fingerprint image. One is capturing from fingerprint scanner, second one is load the external fingerprint image file and the third one is load from database. The fingerprint image is loaded for the original fingerprint image or for the template image. In loading original fingerprint image, user only can get image from fingerprint scanner and external image file. In loading template fingerprint image, user can import the image with all loading approaches. Note that, after loading image, image will be stored in an object called BufferedImage, all processing will execute with this object, so the original image will not be changed during running the system.

### Image pre-processing

The job of this function is that: processing a fingerprint image, make it suitable for extract minutiae and matching. In order to let user can understand of this function, I divided this function into five sub-functions. Each sub-function explains each step in image pre-processing. User can see the result of each sub-function directly. Result of each function will display in the image. In this function, basic data is integer, most calculation based on gray level of the fingerprint image. The main data structure is matrix. Each sub-function will be described in following.

1. Normalization

This function is for making the input image into a format gray level. Some lower quality images’ gray level is not easy for pre-processing (some reason due to it, such as different fingerprint scanner). After this function, an input image can be set up into a format gray level. All the calculation of this function is base on the original average gray level of the image and original variance of the image. This function will not change anything of the fingerprint image (e.g. fingerprint ridge). User will see a gray image after running this function. This function is suitable for image which gray level is too height.

1. Orientation estimation

This function is for calculate the direction of each ridge in the fingerprint image. It provides the direction and prepare for minutiae direction. Calculation of this function is depends on the changing of image gray level. After running this function, user can get an image which include some short line, these short line shows the direction of fingerprint ridge.

1. Edge detection

This function shows the interesting range of a fingerprint image (the useful area of an image). Some images may not be perfect; some area may not be used for calculation. After this function, user can get an image which includes useful area only. After this function, system will throw away the un-useful data and then go to the next stage. Note that this function can be executed with the orientation estimation function at the same time, or it can be executed before orientation estimation.

1. Ridge detection

The job of this function is separation the foreground (ridge) and background. After this function, system provides an image which includes two colors (black and white) only. The image is not gray image after this function. In this function, system will change the image to a binary image; it is helpful for do the following step with a binary image.

1. Thinning

This function is to make fingerprint ridge “thin”. The ridge and edge in the image is too wide, so it is not good for minutiae extraction. This function prepares the image for minutiae extraction; let the result of minutiae extraction more accurate. With this function, user can get a fingerprint image which ridge is thinner.

### Recognition.

This function is the main function of this project. The recognition is minutiae-based in this project. Recognition functional also divided into three sub-functions. User also can see the result of each sub-function.

1. Minutiae extraction

This function is for extracting minutiae. Every minutia will be highlighted in the image. Only two kinds of minutiae are considered in this system, so I use two different colors to highlighted minutiae. In this function, core of an image and delta of an image also will be extracted.

1. Remove false minutiae

After minutiae extraction, user also needs to run this function to remove false minutiae. This function can make the recognition result more accurate.

1. Matching

Matching is a very important functional. This function is for compare two fingerprints. System will return the result to user, this result will not display within image. After this function, system shows the minutia which is the same in two fingerprints in a text area. Note that this system only supports one fingerprint match to another one fingerprint. If I have enough time, I will try to implement one fingerprint match to many other fingerprints.

## Additional functional

Fingerprint management is additional functional of my application. This functional include three sub-functions: fingerprint enrolls, fingerprint update and fingerprint delete. In enroll, user has to do pre-processing and extract minutiae before enroll a fingerprint into database. In update, user cannot change data of fingerprint minutiae. In delete, system will show the select result to user before user delete it, this result include all information (e.g. image, name and ID) of a fingerprint in database.

# Potential users

Today, fingerprint recognition system is widely use. It is for security and personal identification. My fingerprint recognition system is suitable for these ways, so the main user of my system may be the administrator of each department or the member who is working for keep security. I think these two features my system should have:

* Easy for use: I will try my best to design my application’s GUI simpler for using.
* More Accurate: I will try my best to implement each step to get a more accurate result.

# Metrics

* Input Image supportability: This application support BMP type image only.
* Image matching supportability: This application support one image match to another one image, it does not support one-to-many matching.
* Performance: The object of this application is to keep the accurate of the recognition by 70%.