Design and Implementation Biometric Access Control System Using Fingerprint for Restricted Area Based on Gabor Filter

Ashraf El-Sisi
Computer Science Department, Faculty of Computers and Information, Menofyia University, Egypt

Abstract: *Biometric recognition is the use of individual biometric characteristics, such as fingerprint, face, and signature for automatically computerized recognition systems. Fingerprints are the most widely used form of biometric recognition system successfully. However, fingerprint images are rarely of perfect quality. They may be degraded and corrupted due to variations in skin and impression conditions. Thus, image enhancement techniques are employed prior to minutiae extraction to obtain a more reliable estimation of minutiae locations. Minutiae extraction yielded many minutiae resulting from fingerprint irregularities. The aim of this work is the development of a biometric access control system for restricted areas based on individual finger print and Gabor filter for enhancement process. The development system architecture, demonstrating the components, enhancement, minutiae extraction and matching techniques are presented. A software application is written in Matlab and C# to implement algorithms for enhancement, minutiae extraction and matching processing. The resulting minutiae information will be used as a method of identifying matching fingerprints. Also it will be used to register this fingerprint in system database. Finally, verification system, and identification system are implemented. Registration system has facilities namely; automatic registration, manual registration and update for the database to help the administrator to update required information. Based on that processing, an integrated secure system for biometric access control is developed for restricted area with acceptable security level.*

Keywords: *Biometric, fingerprint, enhancement, gabor filter, minutiae extraction, verification, and identification system.*

Received November 6, 2008; accepted May 17, 2009

1. Introduction

Biometric system is automated recognition of persons based on their biological or/and behavioral characteristics. Automated measurement of biological or/and behavioral characteristics of person for medical, security or psychological purposes. Depending on the application context, a biometric system may be called either a verification system or an identification system. A verification system verifies a person by comparing the captured biometric characteristic with his own biometric template pre-stored in the system. It conducts one-to-one comparison to determine whether the identity claimed by the individual is true. A verification system either rejects or accepts the submitted claim of identity.

An identification system recognizes an individual by searching the entire template database for a match. It conducts one-to-many comparisons to establish the identity of the individual. In an identification system, the system establishes a subject's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity. Any human physiological and/or behavioral characteristic can be used as a biometric identifier to recognize a person as long as it satisfies these requirements: Universality, which means that each person should have the biometric; Distinctiveness, which indicates that any two persons should be sufficiently different; Permanence, which means that it should be invariant over a period of time; Collectability, which indicates that it can be measured quantitatively. A number of biometric identifiers are in use in various applications. Each biometric has its strengths and weaknesses and the choice typically depends on the application. No single biometric is expected to effectively meet the requirements of all the applications. In this paper we will focus on the finger print access control application. This type of application can be classified in two verification system or an identification system. The most widely used method for representing a fingerprint is minutiae pattern [15]. The job of minutiae pattern matching is to recognize corresponding minutiae through alignment and pairing. However, it is not easy to extract the minutiae points accurately from the original fingerprint images and the performance of the feature extraction algorithm relies heavily on the quality of the input images. It is essential to implement fingerprint enhancement process before extracting minutiae for the robustness of fingerprint identification algorithm with respect to the quality of fingerprint images. Fingerprint enhancement is intended to reduce noises and improve the contrast between ridges and valleys

in the gray-scale fingerprint images. Much work has been dedicated to fingerprint enhancement and a variety of relevant approaches have been proposed based on contextual filters, Fourier filtering, Gabor filter, and wavelet [1, 6, 7, 13]. According for these approaches, the Gabor filter-based method achieves comparatively favorable performance and is by far the most popular method for fingerprint enhancement [2, 14]. Access control for restricted areas like airports control area, control room of nuclear applications, and dangerous areas of many application are need reliable authenticated access with high level of security for safety. So the goal of this work is the development of a biometric access control system for restricted areas based on individual finger print and good enhancement technique like Gabor filter for improvement process of the fingerprint image with acceptable performance. In the first phase of development Matlab can be used to implement algorithms for enhancement process, minutiae extraction and matching processing. Identifying matching fingerprint used based on extracting minutiae for registration stage, verification stage, and identification stage of developed system. The structure of this paper is as follows. Section 2 gives an overview about fingerprint analysis. In section 3 the details of system level design is presented. Section 4 describes feature extraction and post image processing for our system. The result of the proposed system is shown in section 5. Finally, some conclusions are put forward in section 6.

2. Fingerprint Analysis

A fingerprint is the reproduction of a fingerprint epidermis, produced when a finger is pressed against a smooth surface. The most evident structural characteristic of a fingerprint is a pattern of interleaved ridges and valleys; in a fingerprint image as shown in Figure 1.

Figure 1. Fingerprint structure.

Ridges (also called ridge lines) are dark whereas valleys are bright. Injuries such as superficial burns, abrasions, or cuts do not affect the underlying ridge structure and the original pattern is duplicated in any new skin that grows. Ridges and valleys run in parallel; sometimes they bifurcate and some times they terminate. When analyzed at the global level, the fingerprint pattern exhibits one or more regions where the ridge lines assume distinctive shapes (characterized by high curvature, frequent termination, etc.,). These regions (called singularities or singular regions) may be classified into three typologies: loop, delta, and whorl. Singular regions belonging to loop, delta, and whorl types are typically characterized by $\cap$, Δ and O shapes, respectively. Sometimes whorl singularities are not explicitly introduced because a whorl type can be described in two facing loop singularities. For fingerprints that not contain loop or whorl singularities, it is difficult to define the core. In these cases, the core is usually associated with the point of maximum ridge line curvature. Unfortunately, due to the high variability of fingerprint patterns, it is difficult to reliably locate a registration (core) point in all the fingerprint images. Singular regions are commonly used for fingerprint classification that is, assigning a fingerprint to a class among a set of distinct classes, with the aim of simplifying search and retrieval. At the local level, other important features, called minutiae can be found in the fingerprint patterns. Minutia means small detail; in the context of fingerprints, it refers to various ways that the ridges can be discontinuous. For example, a ridge can suddenly come to an end (termination), or can divide into two ridges (bifurcation). Although several types of minutiae can be considered (the most common types are shown in Figure 2(a), usually only a coarse classification is adopted to deal with the practical difficulty in automatically discerning the different types with high accuracy. The FBI minutiae-coordinate model considers only terminations and bifurcations [8]. Each minutia is denoted by its class, horizontal (x_o) and vertical (y_o) coordinates and the angle between the tangent to the ridge line at the minutia position and the horizontal axis as shown in Figure 2 (b) and (c).

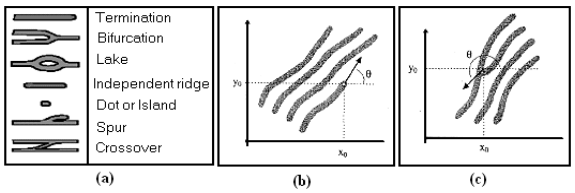


Figure 2. (a) Most common minutiae types (b) A termination minutia (c) A bifurcation minutia.

In practice, an ambiguity exists between termination and bifurcation minutiae; depending on the finger pressure against the surface where the fingerprint is left, terminations may appear as bifurcations and vice versa (this property is known as termination/bifurcation duality). In fact, each ridge of the epidermis (outer skin) is dotted with sweat pores along its entire length and anchored to the dermis (inner skin) by a double row of peglike protuberances, or papillae. Although pore information (number, position, shape, etc.,) is highly distinctive, very few automatic matching techniques use pores since their

reliable detection requires very high resolution and good quality fingerprint images. It is necessary to employ image enhancement techniques prior to minutiae extraction to obtain a more reliable estimate of minutiae locations. The response of a matcher in a fingerprint recognition system is typically a matching score (without loss of generality, ranging in the interval (0, 1)) that quantifies the similarity between the input and the database template representations. A typical biometric verification system commits two types of errors: mistaking biometric measurements from two different fingers to be from the same finger (called false match) and mistaking two biometric measurements from the same finger to be from two different fingers (called false non-match) [9]. Note that these two types of errors are also often denoted as False Acceptance Rate (FAR) and False Rejection Rate (FRR) respectively. Biometric testing literature discusses accuracy in terms of the likelihood of these types of errors. The FRR and FAR for a biometric system always depend on the match threshold and are always inversely related [3]. For a given system, it is not possible to reduce both error rates simultaneously. Depending on the purpose of the system, one type of error might be preferred over the other. The best setting is a result of balancing user convenience (few false rejects) with security objectives (few false accepts), and carefully considering the costs and risks of each error type in context.

3. System Level Design

In this section, we will show the architecture for system development in Figure 3. Through the following subsection each element in this architecture will be described.

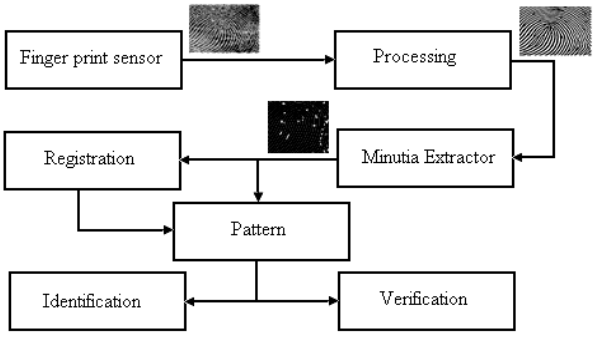


Figure 3. System architecture.

3.1. Fingerprint Sensor

Fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted minutiae points) which is stored and used for matching. Fingerprint sensors are very intricate and continue to grow more complicated. They are becoming a vital part of the transformation to a more technologically integrated society. Current fingerprint technologies are generally susceptible to acquiring poor quality images due to different skin conditions and environmental effects. These poor quality images adversely affect the ability to accurately determine a person's identity. But most of fingerprint sensors can be treat and enhance this poor image and then use it.

3.2. Image Processing and Enhancement

A critical step in automatic fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint images. However, the performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. In order to ensure that the performance of an automatic fingerprint identification/verification system will be robust with respect to the quality of the fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module. This section describes the methods for constructing a series of image enhancement techniques for fingerprint images. These techniques consist of four main stages: normalization, orientation estimation, ridge frequency estimation, and Gabor filtering. In addition to these four stages, three additional stages have been implemented, including segmentation, binarization, and thinning. In this subsection, the methodology for each stage of the enhancement algorithm will be discuss, including any modifications that have been made to the original techniques.

3.2.1. Segmentation

The first step of the fingerprint enhancement algorithm is image segmentation. Segmentation is the process of separating the foreground regions in the image from the background regions. The foreground regions correspond to the clear fingerprint area containing the ridges and valleys, which is the area of interest. The background corresponds to the regions outside the borders of the fingerprint area, which do not contain any valid fingerprint information. When minutiae extraction algorithms are applied to the background regions of an image, it results in the extraction of noisy and false minutiae. Thus, segmentation is employed to discard these background regions, which facilitates the reliable extraction of minutiae. In a fingerprint image, the background regions generally exhibit a very low grey-scale variance value, whereas the foreground regions have a very high variance. Hence, a method based on variance threshold can be used to perform the segmentation. Firstly, the image is divided into blocks and the grey-scale variance is calculated for each block in the image. If the variance is less than the

global threshold, then the block is assigned to be a background region; otherwise, it is assigned to be part of the foreground. The grey-level variance for a block of size WxW is defined as:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=o}^{W-1} (I(i, j) - M(k))^2 \quad (1)$$

where *V(k)* is the variance for block k, *I(i, j)* is the grey-level value at pixel *(i, j)*, and *M(k)* is the mean grey-level value for the block k. The result of segmentation using a variance threshold of 100 can be shown in Figure 4.

(a) Original image. (b) Segmented image.

Figure 4. Result of segmentation using a variance threshold of 100.

3.2.2. Normalization

The next step in the fingerprint enhancement process is image normalization. Normalization is used to standardize the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values. Let *I(i, j)* represent the grey-level value at pixel *(i, j)*, and *N(i, j)* represent the normalized grey-level value at pixel *(i, j)*. The normalized image is defined as:

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{V_0 (I(i,j) - M)^2}{V}} & \text{if} \quad I(i, j) > M, \\ M_0 - \sqrt{\frac{V_0 (I(i,j) - M)^2}{V}} & \text{otherwise} \end{cases} \quad (2)$$

where M and V are the estimated mean and variance of *I(i, j)*, respectively, and *M0* and *V0* are the desired mean and variance values, respectively. Normalization does not change the ridge structures in a fingerprint; it is performed to standardize the dynamic levels of variation in grey-level values, which facilitates the processing of subsequent image enhancement stages. The result of normalization can be shown in Figure 5.

(a) Original image. (b) Normalized image.

Figure 5. The result of normalization.

3.2.3. Orientation Estimation

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. The least mean square estimation method is used to compute the orientation image [6]. However, instead of estimating the orientation block-wise, we have chosen to extend their method into a pixel-wise scheme, which produces a finer and more accurate estimation of the orientation field. The result of estimated orientation process can be shown in Figure 6.

(a) Original image. (b) Estimated orientation.

Figure 6. The result of estimated orientation process.

3.2.4. Ridge Frequency Estimation

In addition to the orientation image, another important parameter that is used in the construction of the Gabor filter is the local ridge frequency. The frequency image represents the local frequency of the ridges in a fingerprint. The first step in the frequency estimation stage is to divide the image into blocks of size W x W. The next step is to project the grey-level values of all the pixels located inside each block along a direction orthogonal to the local ridge orientation. This projection forms an almost sinusoidal-shape wave with the local minimum points corresponding to the ridges in the fingerprint. This involves smoothing the projected waveform using a Gaussian low pass filter of size W x W to reduce the effect of noise in the projection. The ridge spacing *S(i, j)* is then computed by counting the median number of pixels between consecutive minima points in the projected waveform. Hence, the ridge frequency *F(i, j)* for a block centered at pixel *(i, j)* is defined as:

$$F(i, j) = \frac{1}{S(i, j)} \quad (3)$$

given that the fingerprint is scanned at a fixed resolution, then ideally the ridge frequency values should lie within a certain range. However, there are cases where a valid frequency value cannot be reliably obtained from the projection. Examples are when no consecutive peaks can be detected from the projection, and also when minutiae points appear in the block. For the blocks where minutiae points appear, the projected waveform does not produce a well-defined sinusoidal shape wave, which can lead to an inaccurate

estimation of the ridge frequency. Thus, the out of range frequency values are interpolated using values from neighbouring blocks that have a well-defined frequency. The result of ridge frequency estimation process can be shown in Figure 7.

(a) Original image. (b) Ridge frequency estimation result.

Figure 7. The result of ridge frequency estimation process.

3.2.5. Gabor Filter

Once the ridge orientation and ridge frequency information has been determined, these parameters are used to construct the even-symmetric Gabor filter. A two-dimensional Gabor filter consists of a sinusoidal plane wave of a particular orientation and frequency, modulated by a Gaussian envelope. Gabor filters are employed because they have frequency-selective and orientation-selective properties. These properties allow the filter to be tuned to give maximal response to ridges at a specific orientation and frequency in the fingerprint image. Therefore, a properly tuned Gabor filter can be used to effectively preserve the ridge structures while reducing noise. The even-symmetric Gabor filter is the real part of the Gabor function, which is given by a cosine wave modulated by a Gaussian. An even symmetric Gabor filter in the spatial domain is defined as [10]:

$$G(x, y, \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\}\cos(2\pi f x_\theta) \quad (4)$$

$$x_\theta = x\cos\theta + y\sin\theta \quad \text{and} \quad y_\theta = -x\sin\theta + y\cos\theta \quad (5)$$

where θ is the orientation of the Gabor filter, f is the frequency of the cosine wave, σx and σy are the standard deviations of the Gaussian envelope along the x and y axes, respectively, and x_θ and y_θ define the x and y axes of the filter coordinate frame, respectively. The Gabor filter is applied to the fingerprint image by spatially convolving the image with the filter. The convolution of a pixel *(i, j)* in the image requires the corresponding orientation value *O(i, j)* and ridge frequency value *F(i, j)* of that pixel. Hence, the application of the Gabor filter G to obtain the enhanced image E is performed as follows:

$$E(i,j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G(u, v, O(i,j), F(i,j))\, N(i-u, j-v) \quad (6)$$

where O is the orientation image, F is the ridge frequency image, N is the normalized fingerprint image, and w_x and w_y are the width and height of the Gabor filter mask, respectively. The filter bandwidth, which specifies the range of frequency the filter responds to, is determined by the standard deviation parameters σ_x and σ_y. Since the bandwidth of the filter is tuned to match the local ridge frequency, then it can be deduced that the parameter selection of σ_x and σ_y should be related with the ridge frequency. However, in the original algorithm in [3], σ_x and σ_y were empirically set to fixed values of 4.0 and 4.0, respectively. A drawback of using fixed values is that it forces the bandwidth to be constant, which does not take into account the variation that may occur in the values of the ridge frequency. Thus, rather than using fixed values, we have chosen the values of σ_x and σ_y to be a function of the ridge frequency parameter, which is defined as:

$$\sigma_x = k_x F(i,j) \quad \text{and} \quad k_y F(i,j) \quad (7)$$

where F is the ridge frequency image, k_x is a constant variable for σ_x, and k_y is a constant variable for σ_y. This allows a more adaptable approach to be used, as the values of σ_x and σ_y can now be specified adaptively according to the local ridge frequency of the fingerprint image. Furthermore, in the original algorithm, the width and height of the filter mask were both set to fixed values of 11. The filter size controls the spatial extent of the filter, which ideally should be able to accommodate the majority of the useful Gabor waveform information. However, a fixed filter size is not optimal in that it does not allow the accommodation of Gabor waveforms of different sized bandwidths. Hence, to allow the filter size to vary according to the bandwidth of the Gabor waveform, we have set the filter size to be a function of the standard deviation parameters:

$$w_x = 6\sigma_x \quad \text{and} \quad w_y = 6\sigma_y \quad (8)$$

where w_x and w_y are the width and height of the Gabor filter mask, respectively, and σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively. The result of enhancement using Gabor filter can be shown in Figure 8.

(a) Original image. (b) Enhanced image.

Figure 8. Enhancement results using the Gabor filter.

3.2.6. Binarization

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarization is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae. One useful property of the Gabor filter is that it has a DC component of zero, which means the resulting filtered image has a mean pixel value of zero. Hence, straightforward binarization of the image can be performed using a global threshold of zero. The binarization process involves examining the grey-level value of each pixel in the enhanced image, and, if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero. The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

3.2.7. Thinning

The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm is employed, which performs the thinning operation using two sub iterations. This algorithm is accessible in MATLAB via the "thin" operation under the bwmorph function. Each sub iteration begins by examining the neighborhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub iterations continue until no more pixels can be deleted. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonized version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. Figure 9 shows results of applying binarisation and thinning to the enhanced image.

(a) Enhanced image. (b) Binirization image. (c) Thinned image.

Figure 9. Results of applying binarisation and thinning to the enhanced image.

4. Minutiae Extraction and Image Post-Processing

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image. Following the extraction of minutiae, a final image post-processing stage is performed to eliminate false minutiae.

4.1. Minutiae Extraction

The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept [11, 12]. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3x3 window. The CN value is then computed as in equation 9, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighborhood.

$$CN = 0.5 \sum_{i=1}^{8} \left| P_i - P_{i+1} \right| \quad P_9 = P_1 \tag{9}$$

Where Pi is the pixel value in the neighborhood of P. For a pixel P, its eight neighbouring pixels are scanned in an anti-clockwise direction.

Using the properties of the CN as shown in Table 1, the ridge pixel can then be classified as a ridge ending, bifurcation or non-minutiae point.

For example, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation.

Table 1. Properties of the crossing number.

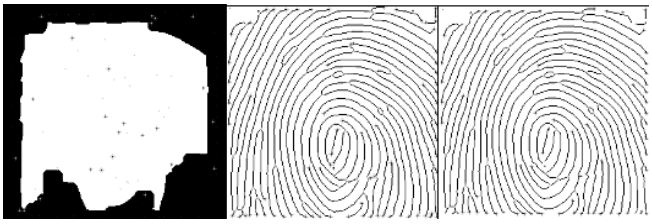
CN	Property
0	Isolated point
1	Ridge ending point
2	Continuing ridge point
3	Bifurcation point
4	Crossing point

After the CN for a ridge pixel has been computed, the pixel can then be classified according to the property of its CN value.

4.2. Post Processing

False minutiae may be introduced into the image due to factors such as noisy images, and image artifacts created by the thinning process. Hence, after the minutiae are extracted, it is necessary to employ a post processing stage in order to validate the minutiae. In order to eliminate false minutiae, the minutiae validation algorithm by Tico and Kuosmanen [12] have chosen to implement. This algorithm tests the validity of each minutiae point by scanning the skeleton image and examining the local neighborhood around the point. The first step in the algorithm is to create an image *M* of size *WxW*, where *M* corresponds

to the *WxW* neighborhood centered on the candidate minutiae point in the skeleton image. The central pixel of *M* corresponds to the minutiae point in the skeleton image, and so this pixel is labeled with a value of -1. The rest of the pixels in *M* are initialized to values of zero. The subsequent steps of the algorithm depend on whether the candidate minutiae point is a ridge ending or a bifurcation. For more details for a ridge ending, and a bifurcation see in [12]. The results of minutia extraction process and removal false minutia can be shown in Figure 10.



(a) Region of interest. (b) Minutia extraction. (c) Remove false minutia.

Figure 10. The results of minutia extraction process and removal false minutia.

4.3. Minutia Match

Given two sets of minutia of two fingerprint images, the minutia match algorithm determines whether the two minutia sets are from the same finger or not. An alignment-based match algorithm is used in our system. It includes two consecutive stages: one is alignment stage and the second is match stage. In the alignment stage, given two fingerprint images to be matched, choose any one minutia from each image; calculate the similarity of the two ridges associated with the two referenced minutia points. If the similarity is larger than a threshold, transform each set of minutia to a new coordination system whose origin is at the referenced point and whose x-axis is coincident with the direction of the referenced point. And in match stage, after we get two set of transformed minutia points, we use the elastic match algorithm to count the matched minutia pairs by assuming two minutia having nearly the same position and direction are identical. The details of two stages are the following:

1. Alignment stage

   Step1: the ridge associated with each minutia is represented as a series of x-coordinates ($x_1, x_2 \ldots x_n$) of the points on the ridge. A point is sampled per ridge length L starting from the minutia point, where the L is the average inter-ridge length. And n is set to 10 unless the total ridge length is less than 10*L. So the similarity of correlating the two ridges is derived from:

$$S = (\sum_{i}^{n} x_i X_i) / (\sum_{i}^{N} x_i X_i)^{1/2} \tag{10}$$

   where ($x_i \ldots x_n$) and ($X_i \ldots X_N$) are the set of minutia for each fingerprint image respectively. And m is minimal one of the n and N value. If the similarity score is larger than 0.8, then go to step 2, otherwise continue to match the next pair of ridges.

   Step2: for each fingerprint, translate and rotate all other minutia with respect to the reference minutia according to the following formula:

$$\begin{pmatrix} x_{i\_new} \\ y_{i\_new} \\ \theta_{i\_new} \end{pmatrix} = TM * \begin{bmatrix} (x_i - x) \\ (y_i - y) \\ (\theta_i - \theta) \end{bmatrix} \tag{11}$$

   where (x, y, θ) is the parameters of the reference minutia, and TM is:

$$TM = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

2. Match stage

   The matching algorithm for the aligned minutia patterns needs to be elastic since the strict match requiring that all parameters (x, y, θ) are the same for two identical minutia is impossible due to the slight deformations and inexact quantization of minutia. We select approach to elastically match minutia is achieved by placing a bounding box around each template minutia. If the minutia to be matched is within the rectangle box and the direction discrepancy between them is very small, then the two minutia are regarded as a matched minutia pair. Each minutia in the template image either has no matched minutia or has only one corresponding minutia. The final match ratio for two fingerprints is the number of total matched pair over the number of minutia of the template fingerprint. The score is 100*ratio and ranges from 0 to 100. If the score is larger than a pre-specified threshold, the two fingerprints are from the same finger.

5. Implementation System and Results

Based on the analysis processing described above sections we build a system to register fingerprint in our database system. Our system is coded by MATLAB for filtering and image processing, and C# for matching and database issues. The experiments were performed on a Pentium Duo Core - 2.0 GHz with 1GB of RAM. The registration mechanism has three options (manual, automatic and update). Verification and identification system is implemented. In verification system, ask the user for his ID then select the fingerprint for this ID from the database in the variable LOD and compare it with the loaded fingerprint. By calling match function, that matches the outside fingerprint with the fingerprint in LOD, this file return the match percentage, compare it with a

defined threshold. Based on this threshold, if percent large than threshold then this finger is verified. Else if percent less than threshold then this fingerprint isn't verified. Otherwise print error message. In identification system we select all the fingerprint images from our database, putting them in variable LOD to compare them with the unknown outside fingerprint, by calling match function that matches the outside fingerprint with every fingerprint in LOD, this file return the match percentage for every pair taking the maximum percentage and then compare it with a defined threshold. Based on this threshold if max_percent large than threshold, then this finger is in the database and return the user's information. Else if max_percent is less than threshold then this fingerprint is not identified. Figure 11 shows screen shot for main menu of system interface for identification.

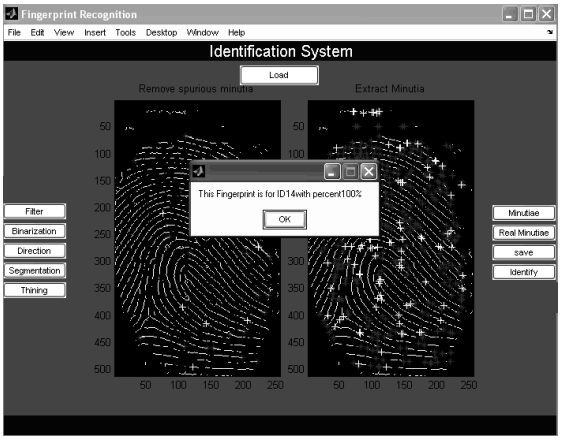


Figure 11. Screen shot for identification.

The majority of the real fingerprint images used for testing was obtained from BioEntry Plus Scanner [10]. Each image is an 8-bit grey-level image scanned at approximately 72 x 72 -dpi resolution and of size 288x288 pixels. In addition to the Scanner data set, tests were also conducted on images from the Fingerprint Verification Competition (FVC2000) database [5]. Our system tests all the images without any fine-tuning for the database. The experiments show our system can differentiate imposturous minutia pairs from genuine minutia pairs in a certain confidence level. For restricted areas application we did not need to cluster database, where in these cases databases are not huge, so we can use complete search in database and decrease the complexity computation required for (verification and identification) systems with acceptable performance. In future to increase the security level by integrate cryptographic into a biometric system [4].

6. Conclusions

A biometric access control system for restricted areas based on individual finger print and Gabor filter for enhancement process is presented in this paper. For fingerprint enhancement, this unique property contributes significantly to improve the image quality. For efficiently we implement image enhancement techniques based Gabor filter. A computer program is coded in Matlab and C# to implement algorithms for enhancement, minutiae extraction and matching processing. The resulting minutiae information was used as a method for identifying matching fingerprints. Registration, verification and identification systems are implemented. Registration system has facilities namely; automatic registration, manual registration and update for the database to help the administrator to update required information. Based on that processing, an integrated secure system for biometric access control is developed for restricted area with acceptable security level. This biometric security system should be open for other, such as face, voice, or eye recognition. To increase security level we can use bimodal (fingerprint and eye). Also we can integrate a biometric system into a cryptographic system.

References

[1] Almansa A. and Lindeberg T., "Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators with Automatic Scale Selection," *Computer Journal of IEEE Transaction Image Process*ing, vol. 9, no. 12, pp. 2027-2042, 2000.

[2] Areekul V., Watchareeruetai U., and Tantaratana S., "Separable Gabor Filter Realization for Fast Fingerprint Enhancement," *in Proceedings of IEEE Internat Conferance on Image Processing*, Italy, pp. 253-256. 2005.

[3] Cappelli R., Maio D., Maltoni D., Wayman J., and Jain A., "Performance Evaluation of Fingerprint Verification Systems," *Computer Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 55-59, 2006.

[4] Chen Y., Chenb L., Tsaia M., Chiu H., Chiu J., and Chon C., "Fingerprint Verification on Medical Image Reporting System," *Computer Journal of Methods Programs in Biomedicine* vol. 89, no. 3, pp. 282-288, 2008.

[5] Fingerprint Verification Database FVC2000, http://bias.csr.unibo.it/fvc, Last Visited 2000.

[6] Hong L., Wan Y., and Jain A., "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *Computer Journal of IEEE Transactions Pattern Annulus Machine Intell*igent, vol. 20, no. 8, pp. 777-789, 1998.

[7] Hsieh T., Lai E., and Wang C., "An Effective Algorithm for Fingerprint Image Enhancement Based on Wavelet Transform," *Computer Journal of Pattern Recognition*, vol. 36, no. 2, pp. 303-312, 2003.

[8] Jain K. and Farrokhnia F., "Unsupervised Texture Segmentation Using Gabor Filters," *Computer Journal of Pattern Recognition*, vol. 24, no. 12, pp. 167-186, 1991.

[9] Maltoni D., Maio D., Jain K., and Prabhakar S., *Handbook of Fingerprint Recognition*, Springer, 2003.

[10] Suprema Inc, www.supremainc.com, Copyright Last Visited 2007.

[11] Thai R., "Fingerprint Image Enhancement and Minutiae Extraction," Technical Report School of Computer Science and Software Engineering, the University of Western Australia, 2003.

[12] Tico M. and Kuosmanen P., "An Algorithm for Fingerprint Image Postprocessing," *in Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, pp. 1735-1739, 2000.

[13] Wei W., Jianwei Li, Huang F., and Feng H., "Design and Implementation of Log-Gabor Filter in Fingerprint Image Enhancement," *Computer Journal of Pattern Recognition Letters*, vol. 29, no. 2, pp. 301-308, 2008.

[14] Yang J., Liu L., Jiang T., and Fan Y., "A Modified Gabor Filter Design Method for Fingerprint Image Enhancement," *Computer Journal of Pattern Recognition Letter*, vol. 24 no. 12, pp. 1805-1817, 2003.

[15] Zhu E., Yin J., and Zhang G., "Fingerprint Matching Based on Global Alignment of Multiple Reference Minutiae," *Computer Journal of Pattern Recognition*, vol. 38 no. 10, pp. 1685-1694, 2005.

Ashraf El-Sisi received the BS and MS in electronic engineering and computer science engineering from Menofyia University, Faculty of Electronic in 1989 and 1995, respectively, and received his PhD in computer engineering & control from Zagazig University, Faculty of Engineering in 2001. His research interest includes privacy preserving data mining, AI approaches in software testing, intelligent Agent, implementation of security and graphics algorithms based on FPGA, testing biometric security algorithms and devices.