Journal of Mobile Multimedia, Vol. 6, No.4 (2010) 314-338
© Rinton Press

FINGERPRINT IMAGE PROCESSING AND FUZZY VAULT IMPLEMENTATION

NANDITA BHATTACHARJEE CHIEN EAO LEE

Monash University, Melbourne

Nandita.Bhattacharjee@infotech.monash.edu.au Celee5@student.monash.edu.au

Received December 16, 2009
Revised May 20, 2010

Accuracy and reliability are two terms that are vital in a biometric system, which must also tolerate the fuzziness of the biometric characteristics to a certain degree. In this paper, we propose and implement fingerprint image enhancement as a preliminary stage to increase the accuracy and reliability of minutiae extraction process for fuzzy vault implementation. In this pre-processing stage, we attempt to recover and enhance the corrupted and noisy region by employing filtering technique. The enhanced image is finally transformed to its skeleton equivalent, preserving the ridges and valleys connectivity for minutiae extraction process. Rutovitz Crossing Number (CN) algorithm is then applied to extract the candidate minutiae which will then undergo a series of minutiae filtering processes to determine the validity of the extracted raw minutiae as true minutia. The implementations of the minutiae filtering processes are able to identify and eliminate the predefined spurious minutiae. As we are focusing on extracting accurate minutiae for the purpose of fuzzy vault implementation, we also take into consideration the quantization of the minutiae, which is an important factor in fuzzy vault locking and unlocking procedures. We then perform the fingerprint fuzzy vault cryptography processes based on the extracted minutiae, where a secret key is generated, encoded and then decoded. Experiments have been conducted for the fingerprint image processing stage and fuzzy vault implementation stage. We obtained a Goodness Index (GI) of 0.55 for the image processing stage, which indicates that our implementation is performing well comparing to other methods. As for the fuzzy vault implementation, we managed to achieve promising False Acceptance Rate (FAR) and False Rejection Rate (FRR) for polynomial degrees ranging from 8 to 13.

Key words: Biometrics, fingerprint, image processing, minutiae extraction, minutiae filtering, fuzzy vault
Communicated by: D. Taniar

1 Introduction

Biometrics is the utilization of physiological or behavioural characteristics in establishing identity [1]. It is undoubtedly one of the most reliable and robust methods of identifying and authenticating an individual, while also ensuring non-repudiation. With the advancement of technology, biometrics has been widely adopted in many areas, ranging from governmental surveillance systems to interactive authentication protocols implemented on personal gadgets. The effectiveness of biometrics systems are based upon the universality, uniqueness, permanence, collectability, performance, acceptability and circumvention of the characteristics used [1]. Research has been widely conducted in efficiently combining biometrics and cryptography, where the uniqueness in biometric templates can replace the function of vulnerable tokens, or analogically, replacing “something one knows” with “something one is”, offering a higher level of authentication certainty.

Among the many available biometric options, fingerprint can be considered as one of the most widely adopted methods in our society. Some of the benefits that fingerprint biometrics offers include its distinguishable uniqueness and collectability. It is also considered as one of the most reliable personal identification methods [2, 3]. However, the efficiency of such systems is highly dependent on the ability of the implemented algorithms to accurately extract the features that can be used for authentication. It is commonly encountered that fingerprint images captured tends to be corrupted with different levels of noise and their qualities often differ due to a wide variety of reasons [4]. As the quality of a fingerprint image remains the major factor in influencing the accuracy of the features extraction process, it is important to employ methods which can filter and recover from the existing noise and corruption, while maintaining the integrity of the fingerprint characteristics and structure. These methods are usually referred to as fingerprint image enhancement or pre-processing stage [4].

On the other hand, it can be deemed almost impossible to obtain identical fingerprint images, even from the same finger when variables such as pressure, orientation, age, skin conditions, etc. would adversely affect the ability of the system to consistently extract the exact identical set of unique features [4]. Therefore, authentication based on exact identical set of fingerprint features data would be unrealistic and impractical. Realizing this fact, research and algorithms have been developed where the fuzziness of fingerprint features data can be tolerated while preserving the integrity of the authentication system. Such fuzziness tolerable system is commonly known as the fuzzy vault [5, 6, 7].

Fuzzy vault is a system that fully utilizes the features information in a fingerprint, which is usually the location of the minutiae in terms of their coordinates and orientation. The logic behind the fuzzy vault scheme is that two fingerprints images can be authenticated if a considerable large amount of their extracted minutiae overlaps, rather than matches exactly in a domain [6]. One of the challenges in implementing fuzzy vault is the ability and effectiveness of the system to establish the valid minutiae from the minutiae extraction and filtering process.

In this paper, we have implemented fingerprint image enhancement algorithm as a pre-processing step to ensure the robustness of the minutiae extraction algorithm with respect to the quality of the fingerprint images. We have also implemented the minutiae extraction algorithm that discovers all the possible minutiae points in the image by employing Rutovitz Crossing Number (CN) concept [4, 8]. Next, we implemented post-processing stages that filters the spurious minutiae detected in the extraction algorithm by employing heuristic rules [9]. Then we quantized the extracted minutiae data with reference to a predefined quantization factor to account for slight variations in minutiae data due to nonlinear distortion, and hence offer standardized minutiae applicable for the fuzzy vault [6]. Finally, we performed the cryptography processes which include encoding and decoding of the secret key based on the extracted minutiae.

The remaining of this paper is structured as follows. Section 2 presents the image enhancement methods or pre-processing stages. Section 3 implements the minutiae extraction whereas Section 4 demonstrates the filtering methods to establish the valid minutiae, which are then being quantized for variation tolerance of fuzzy vault. Finally in Section 5, the fuzzy vault encoding and decoding processes were carried out. Experimental results of our implemented algorithms and conclusions are provided in Section 6 and Section 7 respectively in this paper.

2 Image Pre-processing

The image pre-processing stages that we implemented performed enhancement while transforming the original grey-scale fingerprint image into its corresponding skeleton binary image. The enhancement stages are outlined as shown in Figure 1.

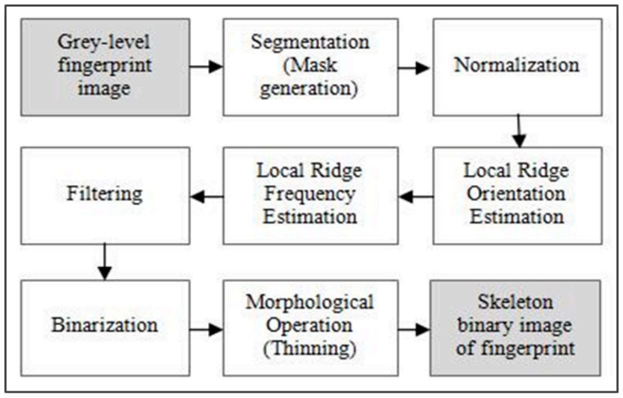

Grey-level fingerprint image
Segmentation (Mask generation)
Normalization
Local Ridge Orientation Estimation
Local Ridge Frequency Estimation
Filtering
Binarization
Morphological Operation (Thinning)
Skeleton binary image of fingerprint

Figure 1 Fingerprint image enhancement process.

2.1 Segmentation

Segmentation is a process to differentiate between the foreground and background regions of a fingerprint image. As foreground regions are usually comprised of alternating ridge and valley structures where useful fingerprint features are present and can be extracted, background regions often carry useless or noisy data. It is important to identify and separate these foreground and background regions in order to avoid extracting features from the background, which could compromise the accuracy of the system, while also induce unnecessary processing time and resources.

We have implemented the segmentation process by comparing the pixel values of the grey-scale fingerprint image with a variance threshold that is pre-specified as follows [10]:

Step 1: *The grey-scale image is divided and processed in blocks of size W x W (W = 16).*

Step 2: *Mean of each block, M(k) is computed using equation (1).*

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

Step 3: *With the corresponding mean values, the variance of each block, V(k) is computed as in equation (2), where I(i, j) is the pixel value at position (i ,j) in the grey-scale image.*

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

Step 4: *The variance of each block is then compared with a threshold value V_{thresh}, which is decided empirically for each different image. If V(k) is higher than V_{thresh}, block k is assigned as foreground; otherwise, it is assigned as background.*

Step 5: *A relative mask of the image is generated by setting the position of foreground pixels with value 1 and background pixels with value 0.*

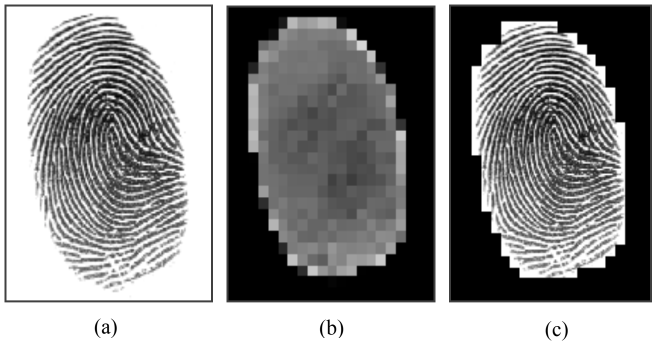


Figure 2 Image segmentation: (a) Grey-scale fingerprint image, (b) Variance image, (c) Segmented fingerprint image.

2.2 Normalization

Normalization is a pixel-wise operation which has no contribution in altering the ridge structure or enhancing the ridge connectivity [4, 11]. Nevertheless, it has the effect of reducing the grey-scale intensity levels and enhancing the contrast between the ridge and valley of the fingerprint image, which remains an important preliminary step for subsequent enhancement stages. The new intensity value for each pixel in the image is redefined based on its previous value and some global parameters as given in equation (3), where I(i, j) and N(i, j) denotes the original and normalized new intensity or grey-scale value at pixel (i, j) respectively. M and V denotes the estimated mean and variance of the image, whereas M_o and V_o are the desired mean and desired variance respectively.

$$N(i,j) = \begin{cases} M_o + \sqrt{\dfrac{V_o(I(i,j) - M)^2}{V}}, & if\ I(i,j) > M \\ M_o - \sqrt{\dfrac{V_o(I(i,j) - M)^2}{V}}, & otherwise \end{cases} \qquad (3)$$

In our paper, we empirically observed that a M_o value of 0 and V_o value of 1 provides a reasonably finest result, with least impact of false ridge disconnection, which is illustrated in Figure 3.

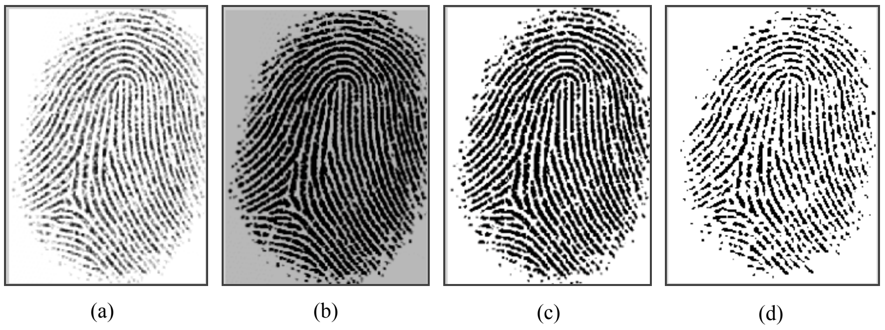


Figure 3 Image normalization: (a) Grey-scale fingerprint image, (b) Normalized image with $M_o = 0$, $V_o = 1$, (c) Normalized image with $M_o = 0$, $V_o = 100$, (d) Normalized image with $M_o = 10$, $V_o = 100$.

2.3 Local Ridge Orientation Estimation

Ridge orientation image is one of the two important data sets required in subsequent filtering process. The local ridge orientation at pixel (i, j) is defined as the angle that the ridge crossing through a small neighbourhood forms with the horizontal axis [8]. We employed the least mean square orientation estimation algorithm developed by Hong, Wan and Jain [11], with major steps briefly summarized as follows:

Step 1: *Normalized image, N is firstly divided into blocks of size W x W (W = 16).*

Step 2: *Horizontal and vertical gradient of each pixel is computed using horizontal and vertical Sobel operators respectively.*

Step 3: *Using the computed gradient and the least mean square estimation method, local orientation of each block is estimated.*

Step 4: *The orientation image is converted into a continuous vector field.*

Step 5: *A two dimensional Gaussian low pass filter is applied to the vector field of the orientation image to correct and smoothen the local ridge orientation image corrupted by noise.*

Step 6: *Finally, the smoothed local ridge orientation estimation at position (i, j) can be computed as shown in equation (4), where O(i, j) is the ridge orientation image, $\Phi'_x(i,j)$ and $\Phi'_y(i,j)$ are the results from the low pass filtering [9].*

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right) \tag{4}$$

2.4 Local Ridge Frequency Estimation

Together with the local ridge orientation image, local ridge frequency is another required data set for the filtering process. The ridge frequency is estimated based on the sinusoidal waveform properties that the ridge and valley in the fingerprint image form. In order to obtain the frequency estimation data, we have implemented the steps provided in [4, 11], which is again briefly summarized as follows:

Step 1: *Normalized image, N is firstly divided into blocks of size W x W (W = 16).*

Step 2: *An oriented window of size L x W (L = 32, W = 16) is constructed for each block centred at pixel (i, j), where oriented means that the window is rotated to have the local ridge orientation of the block aligned with the y-axis.*

Step 3: *The x-signature of the ridges and valleys, X[k] within the oriented window is computed for each block centred at pixel (i, j) using equations (5), (6), and (7).*

$$X[k] = \frac{1}{w}\sum_{d=0}^{W-1} N(u, v), \qquad k = 0, 1, \dots, L-1 \tag{5}$$

where,

$$u = i + \left(d - \frac{W}{2}\right)\cos O(i,j) + \left(k - \frac{1}{2}\right) sin\ O(i,j) \tag{6}$$

$$v = j + \left(d - \frac{W}{2}\right)\cos O(i,j) + \left(k - \frac{1}{2}\right) sin\ O(i,j) \tag{7}$$

The x-signature forms a sinusoidal wave which has the same frequency as that of the ridge and valleys in the oriented window if no singular point or minutia appears in that window. Thus the frequency of the ridges and valleys can be estimated from the x-signature. For such estimation, we first establish the average number of pixels between two consecutive peaks in the x-signature. Then the frequency F(i, j) can be computed as the inverse of average distance between two consecutive peaks of the x-signature. Any window without consecutive peaks in its x-signature will be defined as invalid field for frequency and assigned a value of -1.

Step 4: *For the corrupted blocks or blocks where minutiae or singular point(s) is present, a well-defined sinusoidal wave will not be formed. In this case, interpolations from frequency of neighbouring blocks are required and carried out as in [12].*

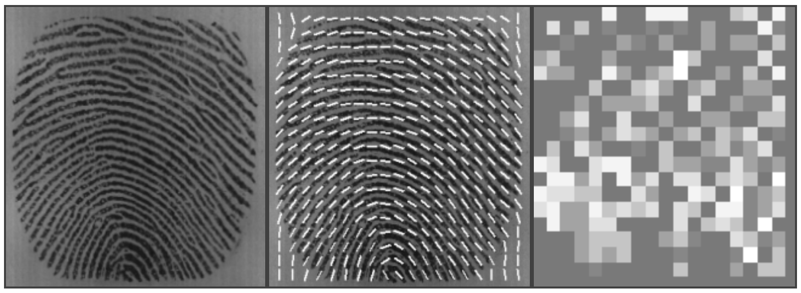


Figure 4 Local ridge orientation and frequency estimation: (a) Grey-scale fingerprint image, (b) Orientation estimation image, (c) Frequency estimation image.

2.5 Filtering

It is common that fingerprint images are corrupted with different levels of noise. Based on the orientation and frequency data computed in previous stages, we defined any data that do not correspond to the relative orientation and frequency as noise. By utilizing the orientation and frequency data, we employed a band pass filter to eliminate the undesired noise, particularly Gabor filter in this paper. The even symmetric Gabor filter can be constructed as follows [4, 11, 13]:

$$G(x, y; \theta, f) = exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\}\cos(2\pi f\, x_\theta), \qquad (8)$$

where,

$$x_\theta = x\sin\theta + y\cos\theta \qquad (9)$$

$$y_\theta = -x\cos\theta + y\sin\theta \qquad (10)$$

and f denotes the frequency of the sinusoidal plane wave from the cosine term in equation (8), σ_x and σ_y represent the standard deviations of the Gaussian envelope along the x and y axes respectively. Among the parameters that are required to be specified in the construction of Gabor filter include θ, f, σ_x and σ_y. While θ and f are determined by the orientation and frequency estimation previously computed, the Gaussian envelope, σ_x and σ_y are both set to 4.0 [11].

As it is computationally inefficient to construct a separate Gabor filter for each of the orientation and frequency data during the filtering process, we can first create and store a bank of filters with equal angle increment from 0 degree to 180 degrees, with each bank for several reasonable frequencies. In this paper, we have constructed and stores 60 filters per bank, each filter with an angle increment of 3 degrees, ranging from 0 degree to 180 degrees, with each bank for the frequencies of 1/5, 1/7, 1/9. Each pixel (i, j) of the image is then convolved with the corresponding filter in such a way that the discrete angle and frequency of the selected Gabor filter from the bank is closest to that of the pixel. The enhanced value of the pixel (i, j), E(i, j) is obtained by the convolution as follows:

$$E(i, j) = \begin{cases} 255, & if\ Mask(i, j) = 0 \\ \displaystyle\sum_{u=-\frac{W_g}{2}}^{\frac{W_g}{2}} \sum_{v=-\frac{W_g}{2}}^{\frac{W_g}{2}} G * N(i-u, j-v), & otherwise \end{cases} \qquad (11)$$

where G = G(u, v; O(i, j), F(i, j)), and W_g is the size of the filter.

2.6 Image Binarization

Although direct minutiae extraction from grey-scale image is possible [14], we implemented image binarization as a preliminary process for minutiae extraction in this paper. It is a stage where the grey-scale values in the enhanced image is transformed and represented in only two levels, which are 0 and 1. The value of 0 and 1, which is usually decided based on threshold comparison, represents a black and white pixel in the binarized image respectively. Several binarization algorithms developed in the past have employed the threshold comparison technique, either locally or globally where the threshold values are determined based on parameters such as orientation, intensity, and others [15]. However, with the application of contextual filtering in the previous sub-section where the local characteristics have been taken into consideration, we can achieve satisfactory binarization outcome B(i, j) by comparing the enhanced image E(i, j) with a global threshold of 0, which is the DC component of the Gabor filter as follows:

$$B(i,j) = \begin{cases} 1, & if\ E(i,j) > 0 \\ 0, & otherwise \end{cases} \tag{12}$$

2.7 Morphological Operation

Instead of utilizing the ridge structure of the fingerprint image for minutiae extraction, we are using the valley image, which is established to be less prone to spurious minutiae [9]. Morphological thinning is an operation that transforms the binarized valley structure of the fingerprint image into a pixel wide, without altering the connectivity of the ridges and valleys provided in the binary image. We applied the mathematical morphology operation developed by Gonzales and Woods [16] to reach the final one-pixel wide version of the fingerprint image. Besides the morphological 'thin' operator, we have also adopted the 'open' and 'fill' operator where the spikes and holes resulting from the intermediate thinning stage are eliminated. These steps are determined empirically to finally obtain a satisfactory outcome.

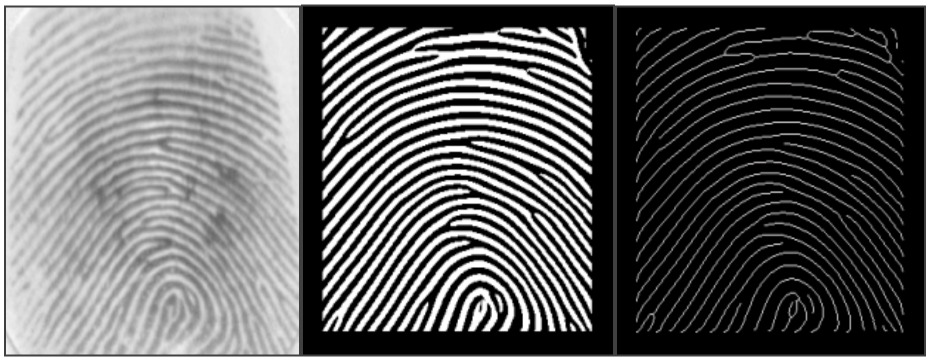

(a) (b) (c)

Figure 5 Filtering, binarization and morphological thinning operations: (a) Grey-scale fingerprint image, (b) Filtered and binarized fingerprint image, (c) Thinned skeleton fingerprint image.

3 Minutiae Extraction

Minutiae points are categorized as level 2 ridge details which correspond to the various ways that ridges can become discontinuous [4]. Several common types of minutiae include ridge ending, bifurcation, island, crossover, etc. In most cases, and also in this paper, we are only interested in considering endings and bifurcations as valid minutiae. Ending is defined as the point where fingerprint ridge abruptly comes to an end, whereas bifurcation is the point where a ridge divides into two ridges [4].

3.1 Rutovitz Crossing Number (CN)

Rutovitz Crossing Number is particularly effective in detecting 4 minutiae types and the continuous ridge pixels in a skeleton fingerprint image [8]. The raw minutiae detection or extraction algorithm using CN can be carried out as follows:

Step 1: *A 3 x 3 block is constructed for each pixel (i, j) that has a value of 1 (skeleton valley) in the skeleton image, where P = P(i, j).*

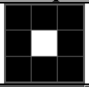
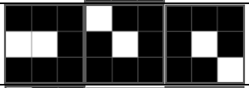
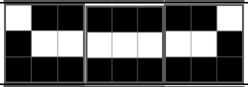
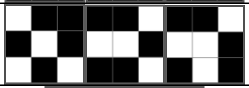
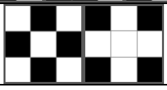
P_1	P_2	P_3
P_8	P	P_4
P_7	P_6	P_5

Step 2: *The CN of each pixel, P(i, j) which is denoted as P in the centre of their corresponding 3 x 3 block is calculated as in equation (13), where the value of P_i, (i = 1 to 8) is either 0 or 1.*

$$CN(P) = \frac{1}{2}\sum_{i=1}^{8}|P_{i\,mod\,8} - P_{i-1}| \tag{13}$$

Step 3: *The minutiae types are determined based on the CN number of each pixel as shown in Table 1.*

Table 1 Crossing number and property

CN(P)	***Property***	***Examples***
0	*Isolation point*	
1	*Ending point*	
2	*Continuing ridge point*	
3	*Bifurcation point*	
4	*Crossing point*	

Step 4: *From the CN that has been computed for each of the valley pixels, which has a value of 1 (or white) in the binary skeleton image, we marked and stored the position or coordinate of the pixel, P(i, j) which has CN(P) of 1 or 3 as candidate endings and candidate bifurcations respectively. These lists of candidate endings and bifurcations minutiae are stored for minutiae filtering process.*

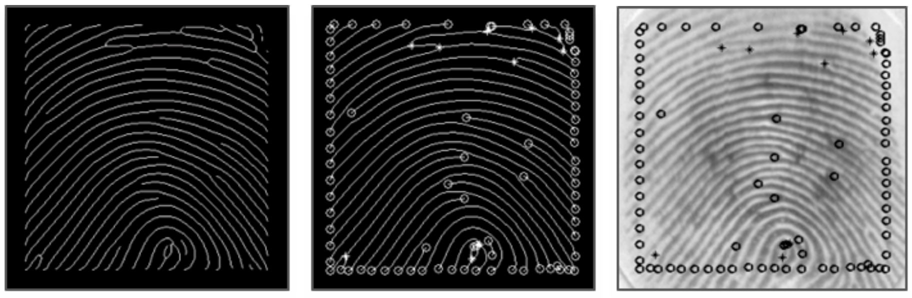


Figure 6 Minutiae extraction using Rutovitz Crossing Number (CN): (a) Skeleton binary fingerprint image, (b) Minutiae extracted from the skeleton image, with 'o' marks the endings and '*' marks the bifurcations, (c) Minutiae extracted on the grey-scale fingerprint image.

4 Minutiae Filtering

Rutovitz Crossing Number (CN) is a definite method where all processed pixels with CN value of 1 and 3 will be considered as listed as endings and bifurcations respectively. While valid minutiae are enlisted, many minutiae formed by breaks, spurs, bridges, borders and other forms of corruption are also included in the minutiae list. Such minutiae are considered as spurious minutiae and must be identified and eliminated from the list.

As the minutiae extraction and filtering processes that we implemented are focused on the fuzzy vault application, it is vital that the minutiae used for locking and unlocking the vault to be as accurate as possible. This is because inaccurate minutiae extraction would adversely affect the False Rejection Rate (FRR) and inefficiency in filtering the spurious minutiae would induce a high False Acceptance Rate (FAR). In this paper, we have implemented algorithms to remove the spurious minutiae at the border, minutiae formed by short breaks, spurs and also the minutiae within a reasonably close distance to each other.

4.1 Border Minutiae Removal

We shall first define the border minutiae as minutiae lying on any of the 4 boundaries of the fingerprint images, which are mostly spurious, and also those positioned within a considerably close distance to the any of the 4 image boundaries, where the ridges are often corrupted with noise, even though they are defined as foreground regions. This is because the regions masking procedure is conducted in block-wise manner. The first step in our minutiae filtering process focuses on removing such border minutiae, which is performed in two stages.

4.1.1 Stage 1: Boundaries Minutiae Removal

The determination and removal of boundaries minutiae is carried out in four directions, which are from the top, bottom, left and right of the boundaries of the image. From the list of raw minutiae extracted previously, we perform the consideration where, if the minutiae position happen to be the first valley pixel (pixel value of 1) from any of the 4 boundaries, it is considered as boundary minutiae and eliminated from the candidate minutiae list.

4.1.2 Stage 2: Near Border Minutiae Removal

In this stage, we apply the windowing technique [17] and threshold comparison method to identify and remove the minutiae that fall into a near-border region as follows:

Step 1: *For each of the minutiae in the candidate minutiae list, a sub-window of size W x W, centred at the minutiae is formed.*

Step 2: *The threshold for comparison is computed as in equation (14).*

$$W_{thresh} = W \, x \, W \tag{14}$$

Step 3: *The sub-window formed in Step 1 is mapped onto the region mask generated previously in the image segmentation process, and the sum, S of the sub-window values in the mask is computed.*

Step 4: *The validity of the candidate minutiae is determined by comparing S and W_{thresh}. If S is equal to W_{thresh}, the minutiae is deemed as valid and remains in the list for further processing, whereas if the value of S is smaller than W_{thresh}, then the minutiae will be deemed as near-border minutiae and eliminated from the candidate minutiae list.*

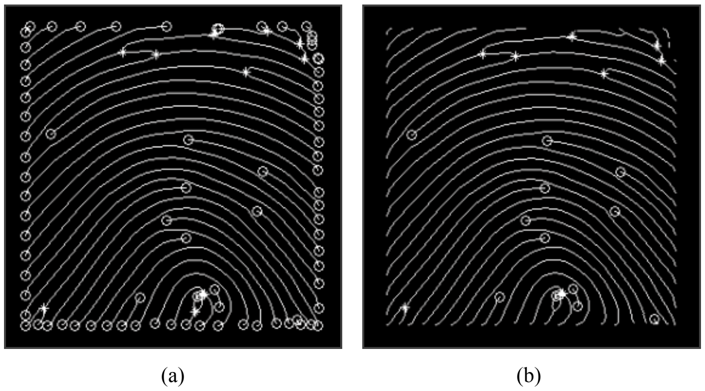


Figure 7 Border minutiae elimination: (a) Raw minutiae detected by CN extraction algorithm, (b) Candidate minutiae after border minutiae removal.

The windowing with threshold comparison technique in Stage 2 is also effective in removing minutiae that lies near the unrecoverable corrupted regions, in which the total mask value of the sub-window will be lower than the threshold, W_{thresh}.

4.2 Short Breaks Removal

Short breaks are the false discontinuity of a continuous ridge, which would be considered as two endpoints by the CN algorithm. Short breaks often occur due to the low quality of or corruption in the fingerprint image. The first attempt to recover from the short breaks or broken ridges was carried out in the image enhancement processes, particularly in the filtering phase. However, it is still common that the attempt would fail to reconnect the broken ridges, leaving them as endpoints. Therefore, it is important to detect such short breaks minutiae in the candidate list, by implementing the following algorithm [9].

If two endpoints satisfy all of the following conditions, they are regarded as short breaks minutiae and will be removed from the candidate list.

Condition 1: The distance between two endpoints is below a threshold, T1.

Condition 2: The difference between the orientation angles of the two endpoints, (θ_A) and (θ_B) is within an interval of [θ_1, θ_2]. The orientation angle of the endpoint is determined by tracing the 8-connected neighbours of the corresponding endpoint.

Condition 3: The difference between the orientation angle of the line connecting the two endpoints, (θ_C) and either (θ_A) or (θ_B) is within an interval of [θ_3, θ_4].

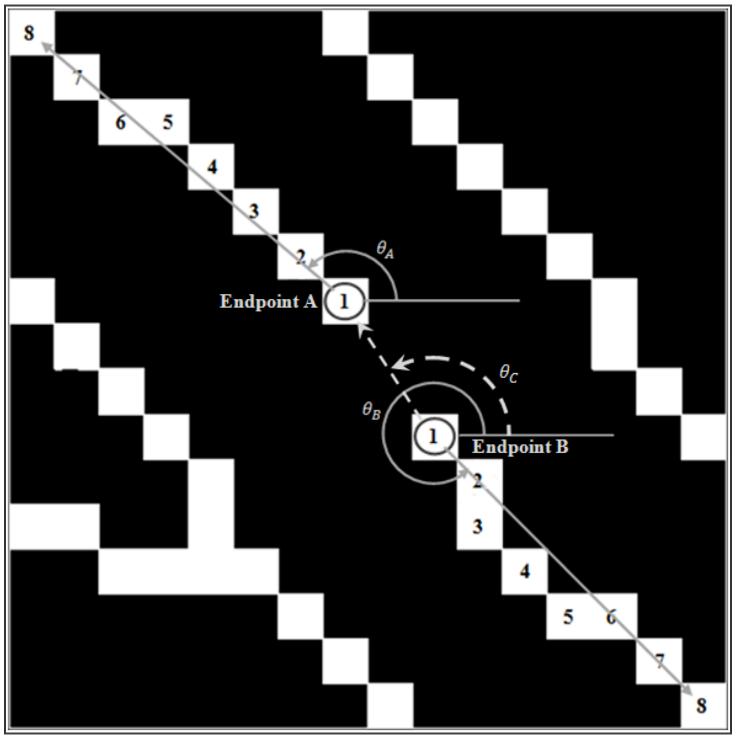


Figure 8 Short break identification parameters illustration.

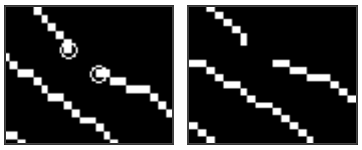


Figure 9 Short breaks detection and removal: (a) Short breaks marked as 2 endpoints by minutiae extraction algorithm, (b) Short breaks minutiae successfully identified and removed.

4.3 Spur Removal

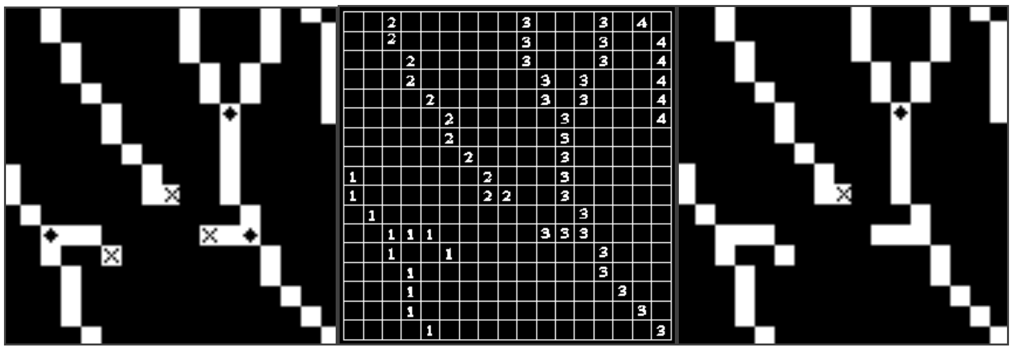
Spur is determined as an endpoint connected to a bifurcation point within a considerably short distance. The algorithms we implemented in identifying and removing spur include windowing and labelling technique [9], with the stages outlined as follows:

Step 1: *The connected pixels in the skeleton fingerprint image are labelled or represented with the same label.*

Step 2: *For each of the pixels with identical labels in the image, check for the presence of endpoint and bifurcation. If such presence is detected, the distance, (d) between the corresponding endpoint and bifurcation having identical label is computed.*

Step 3: *The distance, (d) computed is compared with a predefined threshold, T2. If d is lower than T2, we construct a sub-window, centred at the corresponding bifurcation or endpoint, with the window size of (2d + 1) x (2d + 1).*

Step 4: *Within the sub-window created, we perform the labelling process again for the connected pixels. If the new label provided for the corresponding bifurcation and endpoint are still identical, they are both defined as spur and removed from the candidate minutiae list. Otherwise, they are both preserved.*



(a) (b) (c)

Figure 10 Spur detection and removal: (a) Raw minutiae detected by minutiae extraction algorithm, with endpoints marked as 'x' and bifurcations marked as '♦', (b) Connected pixels are labelled using identical label, (c) 4 minutiae detected due to spur are removed by spur removal algorithm, whereas the other 2 remain as candidate minutiae.

4.4 Close Distance Minutiae Removal

This stage of minutiae filtering removes minutiae that are positioned within a close distance to each other. For each of the minutiae in the candidate list, we construct a sub-window centred at the minutiae position. The threshold of distance tolerance, T3 would decide on the sub-window size, which is computed as ((T3 + 1) x (T3 + 1)). In this paper, we establish the threshold as the distance between two parallel and neighbouring valleys, where the minutiae is on. We used the term valley instead of ridge because the output of our pre-processing stages is the skeleton of the fingerprint valleys. Identification of close distance minutiae is then carried out in each sub-window generated. If there is more than one minutia within the sub-window, all the minutiae in the sub-window would be considered as close distance minutiae and eliminated from the candidate minutiae list. This method is also effective in eliminating spurious minutiae caused by bridge structure formed between two neighbouring valleys (or ridges). After the close distance minutiae removal stage, we considered the minutiae that remain in the candidate minutiae list as valid minutiae. We refer to the minutiae list consisting of all the surviving minutiae as valid minutiae list, which will be used in the fuzzy vault authentication process.

4.5 Quantization of Minutiae Position

Prior to the locking and unlocking process of the fuzzy vault with the minutiae in the valid minutiae list, we quantized the values given by each minutiae in terms of their two dimensional coordinate position in the fingerprint image. A quantization factor is to be decided, which will be an element in deciding the tolerance and sensitivity of the vault. A quantization factor that is too large would increase the tolerance, or more specifically, the False Acceptance Rate (FAR), whereas a quantization that is too low would have a high sensitivity that might not successfully account for the fuzziness in fingerprint, inducing a high False Rejection Rate (FRR) in the authentication process.

Tessellated blocks of $\mathbf{Q} \times \mathbf{Q}$ is formed in the fingerprint image, where $\mathbf{Q}$ is the selected quantization factor. Every minutia that falls within the corresponding tessellated space formed will be quantized as follows:

$$X_Q = ceil\left(\frac{X_M}{Q}\right) x\, Q - floor\left(\frac{Q}{2}\right) \tag{15}$$

$$Y_Q = ceil\left(\frac{Y_M}{Q}\right) x\, Q - floor\left(\frac{Q}{2}\right) \tag{16}$$

where X_Q, Y_Q are the quantized X and Y coordinate of the minutiae whereas X_M, Y_M are the un-quantized X and Y coordinate as provided in the valid minutiae list respectively. As illustrated in Figure 11, every minutiae positioned within a tessellated space quantized with Q = 7 will be considered as having the centred position in its specified window, or analogically, we could consider only one standardized coordinate for every pixel within each tessellated block.

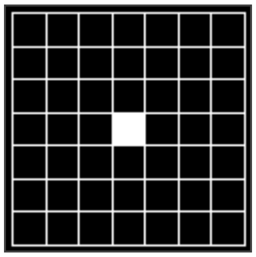


Figure 11 Quantization within tessellated space of quantization factor, Q (Q = 7), where every pixel will be regarded as having the centred position as marked.

With the quantized valid minutiae list, we have finally constructed a locking minutiae list. A similar procedure is also performed for the candidate unlocking minutiae list, which will later undergo an alignment process before being evaluated for authentication with the locking list in the fuzzy vault.

5 Fuzzy Vault

Key is an object that is often linked with authentication. The utilization of key has enabled many encryption and decryption methods to be developed to provide strong cryptography mechanisms. Nevertheless, while depending on the secret key, many methods overlooked the importance of key protection, which has become the vulnerable point of attack in a cryptosystem. Realizing the shortcomings of conventional key protection methods, which often rely on attackable user-selected passwords, fuzzy vault implementations where biometric features used as key protection mechanism has emerged as a favourable concept in cryptography.

5.1 Encoding

Utilizing the secret key and minutiae points extracted, the fuzzy vault encoding can be performed, as illustrated in Figure 12.

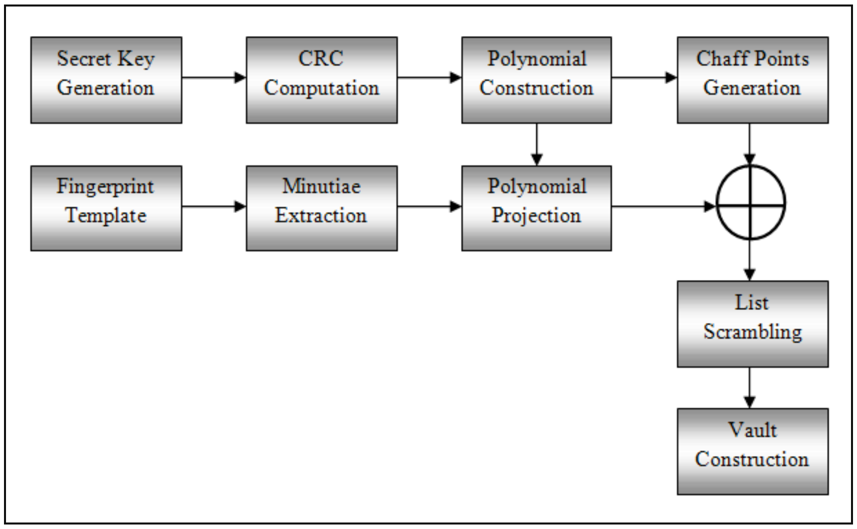
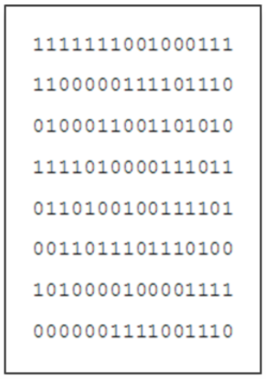


Figure 12 Block diagram for fuzzy vault encoding process.

5.1.1 Secret Key Generation

In fuzzy vault implementation, a secret key, denoted as K is firstly generated. K is comprised of fixed length random bits generated by the system to be bound with the biometric template presented during the encoding or vault locking procedure. For a field of GF(2^{16}), the length of K generated is in a multiple of 16-bits. Figure 13 depicts an example of 128-bits key randomly generated by the fuzzy vault encoding system.



```
1111111001000111
1100000111101110
0100011001101010
1111010000111011
0110100100111101
0011011101110100
1010000100001111
0000001111001110
```

Figure 13 An example of 128-bits key generated for fuzzy vault encoding purpose.

5.1.2 Cyclic Redundancy Check (CRC) Computation

CRC is an error detecting code commonly employed in noisy communication systems where the checksum is computed and appended to the message being transmitted. Reference to the checksum enables the receiver to verify the correctness of the received data. In this fuzzy vault implementation, a 16-bits CRC is generated with regard to the generated secret key, K.

The 16-bits primitive polynomial, particularly ‘CRC-16’ as provided in equation (17) is employed for CRC computation. The generated 16-bits CRC is then appended to K. Thus, for a 128-bits key, a stream of 144-bits data, denoted as K|CRC is computed.

$$G_{CRC}(x) = x^{16} + x^{15} + x^2 + 1 \tag{17}$$

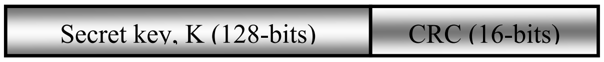


Figure 14 The 128-bits secret key concatenated with the corresponding 16-bits CRC.

5.1.3 Polynomial Construction

In order to construct the encoding polynomial, the secret key, K is segmented as represented as coefficients. For encoding in GF(2^{16}), each coefficient would be comprised of 16-bits in order to be represented as an element in the Galois Field. The data stream, K|CRC of 16 (D + 1) bits is partitioned into (D + 1) non-overlapping 16-bits coefficients, represented as C_0, C_1, …, C_D. With the coefficients computed, the encoding polynomial, P can be constructed as shown in equation (18) where the variable u is obtained by concatenating the x and y coordinate of the locking minutia (u = (x|y)).

$$P(u) = C_D u^D + C_{D-1} u^{D-1} + \cdots + C_1 u + C_0 \tag{18}$$

5.1.4 Polynomial Projection

Polynomial projection is a process that convolves the constructed polynomial, P with the r minutiae used for the locking procedure to conceal the secret key, K. In order to perform that, P is evaluated at all of the r selected minutiae points, as $P(u) = \{P(u_j)\}_{j=1}^{r}$ to form the locking set, G where $G = \{(u_j, P(u_j))\}_{j=1}^{r}$ and u_j is a 16-bits element representing the two-dimensional position of a minutia.

5.1.5 *Chaff Points Generation*

Chaff points are generated in order to hide the genuine template from attackers that happen to gain access to the vault. However, the amount and position of chaff points to be introduced into the vault is restricted by the variance in the fingerprint capturing and feature extraction algorithm. The distance of the chaff points must not be too close in order to account for the quantization procedure. By deciding on a distance threshold (d), chaff points can be positioned in any location as long as their distances from all genuine points satisfy the threshold. In addition, placing a chaff point within the threshold distance from the genuine point would immediately reveal to the attacker that the chaff points are indeed not genuine.

For a chaff set with an amount of s chaff points, a number of s random points, z_1, z_2, …, z_s are firstly generated in GF(2^{16}), with the constraint that none of the random points overlap with the genuine minutia points, u. This relationship can be outlined as $z_i \neq u_j$, where i = 1, 2, …, s and j = 1, 2, …, r. Next, another set of s chaff points, $z'_1, z'_2, \ldots z'_s$ is generated, with the constraint that the combination of the pairs (z_i, z'_i), where i = 1, 2, …, s do not fall onto the constructed polynomial, P. The chaff set can then be represented as Z, where $Z = \{(z_k, z'_k)\}_{k=1}^{s}$.

5.1.6 *List Scrambling and Vault Construction*

The list scrambling is firstly carried out as $G \cup Z$, which is the union of the genuine set, G and the chaff set, Z. Then the union list is randomized with the purpose of removing any stray information that may cause the discovery and lead to the separation of genuine points from the chaff points. The scrambled list then constitutes the vault, which is represented as $V = \{(v_i, w_i)\}_{i=1}^{T}$, where T = r + s, is the total number of points in the vault.

5.2 Decoding

During unlocking attempts, minutia points from the enquiry fingerprint will be extracted and presented for the decoding process, which is illustrated in Figure 15.

5.2.1 *Candidate Points Identification*

Candidate points for unlocking purpose are identified by comparing the points in the vault, V with the aligned list of unlocking points submitted. The list of submitted points is denoted as E which consists of a number of N minutia points, $u'_1, u'_2, \ldots, u'_N$. The candidate points for vault unlocking purpose are selected in such a way that if u'_i, (i = 1, 2, …, N) matches with any of the abscissa values of V $(V = \{(v_i, w_i)\}_{i=1}^{T})$, namely v_j, (j = 1, 2, …, T), the corresponding vault point, (v_j, w_j) is enlisted as the point to be used for the subsequent polynomial reconstruction process. The composed list of points for the polynomial reconstruction is denoted as L'.

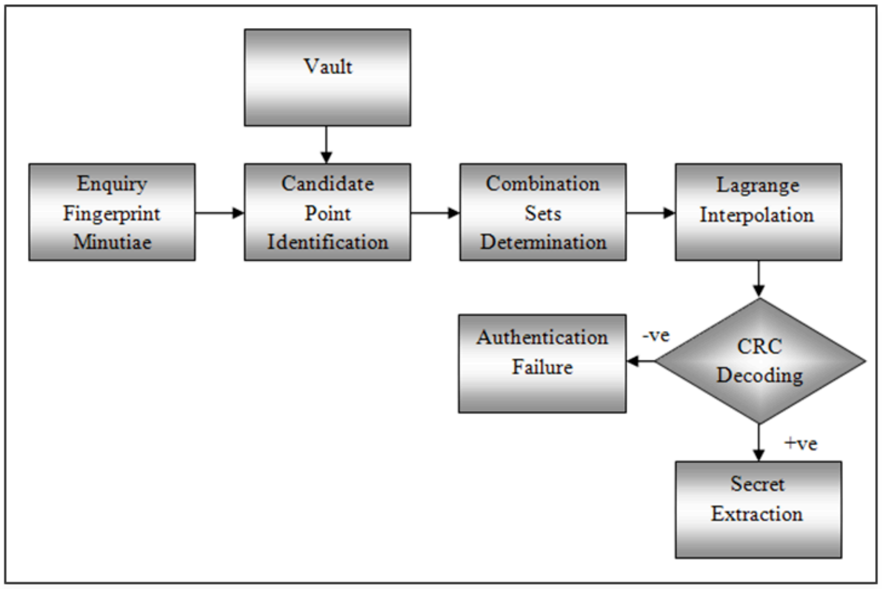

Vault
Enquiry Fingerprint Minutiae
Candidate Point Identification
Combination Sets Determination
Lagrange Interpolation
Authentication Failure
-ve
CRC Decoding
+ve
Secret Extraction

Figure 15 Block diagram for fuzzy vault decoding process.

5.2.2 Combination Sets Determination

From the composed list L', possible sets are computed for the polynomial reconstruction attempts. In order to reconstruct a polynomial of D degrees using polynomial interpolation method, (D + 1) unique projections are required. The unlocking process will be cancelled and considered as a failure if the number of points in the list L' is less than (D + 1). Otherwise, all possible sets, each with different combinations of (D + 1) points from L' will be used for the polynomial reconstruction attempts. The total number of different attempt sets, $\text{Total}_{\text{ATP}}$ that can be composed for the L' with F listed points is determined as equation (19).

$$Total_{ATP} = \frac{F!}{(D+1)!\,(F-(D+1))!} \tag{19}$$

5.2.3 Lagrange Polynomial Interpolation

Lagrange polynomial interpolation is a form of numerical analysis where provided with a set of points, the polynomial that intersects at every provided point is constructed. For the fuzzy vault unlocking purpose, the encoding polynomial for the list L', where $L' = \{(v_1, w_1), (v_2, w_2), \ldots, (v_{D+1}, w_{D+1})\}$ is reconstructed as shown in equation (20). The resulting polynomial is then represented as in equation (21).

$$P'(u) = \frac{(u - v_2)(u - v_3) \dots (u - v_{D+1})}{(v_1 - v_2)(v_1 - v_3) \dots (v_1 - v_{D+1})} w_1 + \frac{(u - v_1)(u - v_3) \dots (u - v_{D+1})}{(v_2 - v_1)(v_2 - v_3) \dots (v_2 - v_{D+1})} w_2 + \cdots + \frac{(u - v_1)(u - v_2) \dots (u - v_D)}{(v_{D+1} - v_2)(v_{D+1} - v_3) \dots (v_{D+1} - v_D)} w_{D+1} \quad (20)$$

$$P'(u) = C'_D u^D + C'_{D-1} u^{D-1} + \cdots + C'_1 u + C'_0 \quad (21)$$

5.2.4 Cyclic Redundancy Check Decoding

From the polynomial in equation (21), the coefficients are concatenated and mapped back to the form of K|CRC where K represents the secret key and CRC is the corresponding checksum for K. In CRC decoding process, the polynomial corresponding to K|CRC is then divided with the primitive polynomial, $G_{CRC}(x) = x^{16} + x^{15} + x^2 + 1$ for the purpose of determining if error has occurred in K that is reconstructed from the polynomial. A non-zero remainder division result indicates the presence of error where incorrect secret key, K has been decoded, whereas a zero remainder division result verifies the decoded K as the actual secret key in the encoding process.

5.2.5 Secret Key Extraction

A satisfactory CRC decoding yields a successful authentication procedure where the CRC codes are removed from the K|CRC formed during polynomial reconstruction. The resulting sequence is released as the secret key, K.

6 Experimental Evaluations and Results

In order to evaluate the performance of our implemented algorithms for image processing, we carried out a series of experiments using the fingerprint images in database FVC2000, FVC2002 and FVC2004. We have performed the evaluation on our minutiae extraction and filtering algorithms after out pre-processing stage based on the goodness index (GI) measurement [18]. Specifically, we compared the valid extracted minutiae that survived the filtering process and with the ground truth minutiae that are identified by human expert. The GI that we computed is based on the formula as shown in equation (22), where P represents the total number of paired minutiae and T represents the ground truth minutiae in a given fingerprint image. A and B are the missed and spurious minutiae respectively whereas U is the total detected minutiae.

$$GI = \left(\frac{P}{T}\right) - \left(\frac{A + B}{U}\right) \quad (22)$$

In this paper, paired minutiae refer to any extracted minutiae that lies within the 8 x 8 tolerance window with its counterpart in the ground truth minutiae marked. Missed minutiae are the ground truth minutiae that are not extracted by the minutiae extraction or did not survive the filtering process, whereas spurious minutiae refer to the extracted minutiae that are not matched with the ground truth minutiae. A high GI, with a maximum value of 1 indicates a high accuracy of the minutiae extraction and filtering algorithm, where most of the minutiae are correctly matched. In our experimental trials, we have implemented the threshold values as given in Table 2. We then present the GI values computed for a set of 10 fingerprint images obtained from FVC2000, FVC2002 and FVC2004 database in Table 3.

Table 2 Threshold values used in experiments.

Threshold Applications	*Threshold Name*	*Threshold Value*
Border Minutiae Removal	W_{thresh}	15
Short Breaks Removal	$T1$	9
Short Breaks Removal	θ_1	125^o
Short Breaks Removal	θ_2	225^o
Short Breaks Removal	θ_3	-25^o
Short Breaks Removal	θ_4	25^o
Spur Removal	$T2$	6
Close Distance Minutiae Removal	$T3$	Determined by algorithm
Quantization	Q	7

Table 3 GI values for a set of 10 fingerprint images.

Fingerprint Image	*P*	*T*	*A*	*B*	*U*	*GI*
1	11	12	1	1	12	0.75
2	14	17	3	2	16	0.51
3	15	19	4	0	15	0.52
4	21	25	4	2	23	0.58
5	26	30	4	6	32	0.55
6	23	28	5	8	31	0.40
7	30	35	5	6	36	0.55
8	35	38	3	3	38	0.76
9	24	27	3	6	30	0.59
10	17	21	4	9	26	0.31

Based on the results we have obtained by using the GI measurement on 10 random fingerprint images in the database, we manage to acquire GI values ranging from 0.31 to 0.75 and average GI of

0.55, which shows a promising reliability in extracting the true minutiae from fingerprint images. The average GI value obtained is comparable to the corresponding values obtained in [19] and outperformed the results presented by several other implementations [9, 11, 20] as shown in Table 4.

Beside the GI values, we evaluated our overall implementation which include the image enhancement, minutiae extraction and filtering in terms of average error rates as provided in Table 5. The average error rates are computed in terms of 3 categories, including the missed, false and type exchanged minutiae with respect to the detected minutiae. From our evaluation result, it is evident that our implementation can achieve better results in terms of false minutiae and total error rates comparing to the implementation results of [14, 21, 22] as reported in [9].

Table 4 GI values for different processing methods.

Methods	*min(GI)*	*max(GI)*	*Average(GI)*
Ratha et al. [20]	-2.38	-1.19	-1.65
Hong et al. [11]	0.29	0.55	0.39
Zhao et al. [9]	0.18	0.75	0.50
Proposed	0.31	0.75	0.55
Simon-Zorita et al.[19]	0.33	0.76	0.55

Table 5 Average error rates of minutiae extraction.

Methods	*Missed (%)*	*False (%)*	*Type-exchanged (%)*	*Total Error (%)*
Proposed	13.9	8.1	8.5	30.5
Maio et al. [14]	6.5	11.8	13.1	31.4
Cheng et al.[21]	15.9	9.6	10.4	35.9
Kim et al.[22]	13.8	25.8	6.3	45.9

We also exhibit the outcomes of the minutiae extraction and filtering process after implementation of our pre-processing stages graphically. In Figure 16, we present two sets of original fingerprint images from the same finger and their thinned images with survived minutiae from the filtering processes marked. The original images with minutiae marked accordingly are also illustrated. From the 2 sets of images, we observe that the surviving minutiae are actually true minutiae that we can manually identify in the images.

Besides minutiae extraction and filtering, we also conducted the experiments of encoding and decoding in fuzzy vault implementation using the minutiae extraction results [6]. In this paper, we consider a fuzzy vault implementation where a 128-bits key is represented as polynomial coefficients and later encoded using the locking minutiae set extracted from our algorithms. In unlocking procedure, another set of minutiae extracted from enquiry fingerprint is passed to the fuzzy vault authentication algorithm.

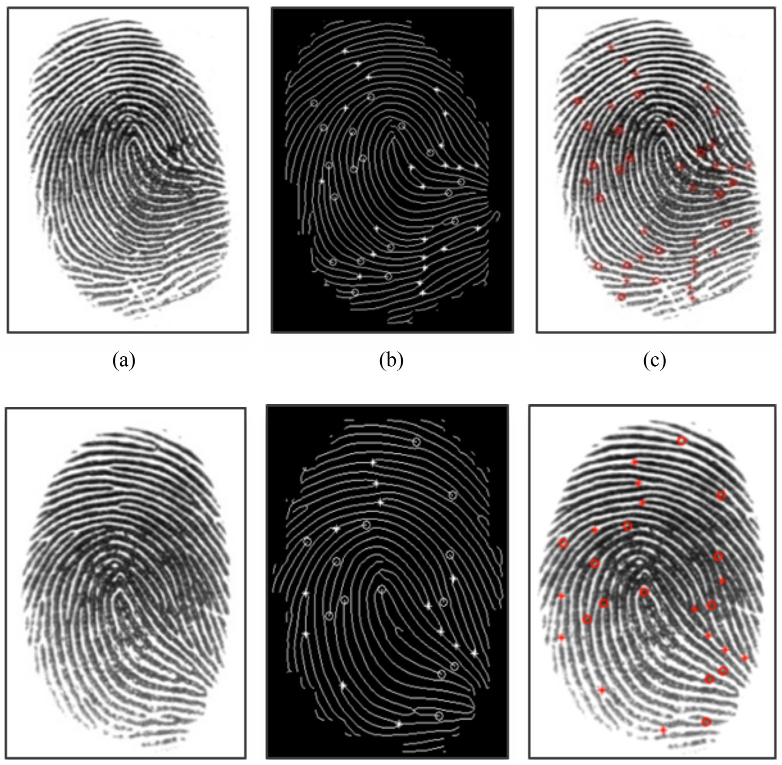


Figure 16 Example of image pre-processing and minutiae extraction result: (a) and (d) Original fingerprint images, (b) and (e) Thinned fingerprint images with extracted minutiae marked, (c) and (f) Original fingerprint images with minutiae marked.

With a D degrees polynomial, minimum of (D + 1) unique matching minutiae are required for polynomial projection to successfully unlock the vault using Lagrange polynomial interpolation [6]. We have performed the evaluation on our fuzzy vault based on FVC2002 fingerprint database. Each database consists of 100 sets of fingerprints with 8 imprints from different finger. However, the evaluation is carried out by using only 4 out of 8 impressions from each finger, based on the considerations determined in [23, 24]. We have also implemented the distance and orientation comparison method for minutiae alignment purpose which does not require extra information other than the minutiae coordinate [24].

For our fuzzy vault implementation, 20 genuine points are used as locking set and 200 chaff points are added into the vault. Fingerprint images where less than 20 minutiae are extracted will be regarded as failure to capture. We have employed Lagrange polynomial interpolation as our polynomial reconstruction method, where if at least (D + 1) minutiae can be matched between the locking and unlocking minutiae set, the polynomial can be reconstructed. The performance of fuzzy vault is evaluated in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR). Another important parameter that has been taken into account while performing the fuzzy vault implementation evaluation is the degree of the encoding polynomial, which served as a major factor in affecting the error rates.

$$FRR = \frac{\sum false\ rejection(s)}{\sum authentication\ attempt(s)} \tag{23}$$

$$FAR = \frac{\sum false\ acceptance(s)}{\sum authentication\ attempt(s)} \tag{24}$$

The FRR evaluation is carried out where each fingerprint is matched against the remaining sample of the same finger during the attempts to unlock the vault and retrieve the key. While one impression is used as enrolment fingerprint, the remaining three impressions are used as enquiry fingerprints. For FAR evaluation, the first impression of each fingerprint identity is provided as the enrolment fingerprint whereas first impression of all remaining fingerprint identities are submitted as enquiry fingerprints for vault unlocking attempts. The evaluations are performed for encoding polynomial of 8 to 13 degrees, each degree with their corresponding key length. We present our experiment results in terms of FRR and FAR as given in Table 6.

Table 6 Experiment results of fuzzy vault authentication.

	FVC2002 DB1	
	FAR (%)	***FRR (%)***
Degree 8 polynomial	0.42	6.25
Degree 9 polynomial	0.15	10.83
Degree 10 polynomial	0.09	12.87
Degree 11 polynomial	0.06	14.95
Degree 12 polynomial	0.02	18.82
Degree 13 polynomial	0.01	25.96

The results in Table 6 indicates the practicality of the algorithms in our minutiae extraction and filtering algorithm after the implemented pre-processing stages for fuzzy vault employment. The outcomes obtained shows that the FRR for the polynomial of degrees 8 to 11 provide better results than those presented in [24] after the pre-processing and post-processing stages for minutiae extraction.

7 Conclusions

Biometrics authentication method is extremely reliable due to its resistance to forgery of the unique characteristics. However, it is also important that such uniqueness can be extracted accurately and utilized efficiently. In fingerprint-based authentication system, one of the major requirements is the tolerance to fuzziness while maintaining the accuracy. In this paper, we have implemented the proposed image pre-processing steps that are considered as important preliminary stages for accurate minutiae extraction. We have also applied the minutiae filtering as post-processing steps to eliminate spurious minutiae. The valid minutiae are then quantized as input for fuzzy vault locking and unlocking procedures. Finally, we have implemented the fingerprint fuzzy vault utilizing the extracted minutiae data. Based on the experimental results, we find that such combination of processes enable us to achieve promising results in terms of viability in fingerprint fuzzy vault implementation.

Acknowledgements

The authors wish to express their gratitude towards Biometric System Laboratory for allowing access to the FVC2000, FVC2002 and FVC2004 database of fingerprints.

References

1. Arun A. Ross, K. Nandakumar, and A. K. Jain, Handbook of Multibiometrics, Springer, 2006.
2. Federal Bureau of Investigation, The Science of Fingerprints: Classification and Uses, U.S. Government Printing Office, Washington, D.C. 1984.
3. Henry C. Lee, R. E. Gaensslen, editors, Advances in Fingerprint Technology, Elsevier, New York, 1991.
4. D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, Springer, 2009.
5. A. Juels, and M. Sudan, A Fuzzy Vault Scheme, Proc. IEEE Int'l Symp. On Information Theory, Lausanne, Switzerland, 2002, pp. 408.
6. U. Uludag, S. Pankanti, and A. Jain, Fuzzy Vault for Fingerprints, Proc. Audio and Video-based Biometric Person Authentication, New York, USA, 2005, pp. 310 – 319.
7. A. Juels and M. Wattenberg, A Fuzzy Commitment Scheme, Proc. 6th ACM Conf. On Computer and Communications Security, Kent Ridge Digital Labs, Singapore, 1999, pp. 28 – 36.
8. D. Rutovitz, Pattern Recognition, J. Roy. Stat. Soc. 129, 1996, pp. 504 – 530.
9. F. Zhao and X. Tang, Preprocessing and Postprocessing for Skeleton-based Fingerprint Minutiae Extraction, Pattern Recognition, Vol. 40, No. 4, 2007, pp. 1270 – 1281.
10. Mehtre and Chatterjee, Segmentation of Fingerprint Images – A Composite Method, Pattern Recognition, Vol. 22, No. 4, 1989, pp. 381 – 185.
11. L. Hong, Y. Wan and A. K. Jain, Fingerprint Image Enhancement: Algorithm and Performance Evaluation, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, 1998, pp. 777 – 789.
12. G. Sapiro and A. Bruckstein, A B-Spline Based Affine Invariant Multi-scale Shape Representation, Proc. 7th Int'l Conf. Image Analysis and Processing 3, Monopoli, Italy, 1993, pp. 20 – 22.
13. A. K. Jain and N. K. Ratha, Object Detection Using Gabor Filters, Pattern Recognition, Vol. 30, No. 2, 1997, pp. 295 – 309.
14. D. Maio and D. Maltoni, Direct Gray-scale Minutiae Detection in Fingerprints, IEEE Transaction PAMI 19 (1997) 27.
15. Q. Xiao and H. Raafat, Fingerprint Image Postprocessing: A Combined Statistical and Structural Approach, Pattern Recognition, 24(10). 1991, pp. 985 – 992.
16. R. C. Gonzalez, R. E. Woods, S. L. Eddins, Digital Image Processing using Matlab, 2nd Edition, Gatesmark Publishing, 2009.

17. A. Tariq, M. U. Akram, S. Nasir and R. Arshad, Fingerprint Image Postprocessing Using Windowing Technique, The Int'l Conf. on Image Analysis and Recognition (ICIAR08), Portugal, 2008.
18. Y. Chen, S. Dass, A. K. Jain, Fingerprint Quality Indices for Predicting Authentication Performance, AVBPA, 2005, pp. 160 – 170.
19. D. Simon-Zorita, J. Ortega-Garcia, S. Cruz-Llanas and J. Gonzalez-Rodriguez, Minutiae Extraction Scheme for Fingerprint Recognition Systems, Proc. International Conference on Image Processing, Vol. 3, 2001, pp. 254 – 257.
20. N. Ratha, S. Chen, and A. K. Jain, Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images, Pattern Recognition 28 (Nov), pp. 1627 – 1672.
21. J. Cheng, and J. Tian, Fingerprint Enhancement with Dyadic Scale-space, Pattern Recognition Lett. 25 (11), 2004m pp. 1273 – 1284.
22. S. Kim, D. Lee, and J. Kim, Algorithm for Detection and Elimination of False Minutiae in Fingerprint Images, Proc. 3rd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA '01), Halmstad, Sweden, 2001, pp. 235 – 240.
23. D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, FVC2002: Second Fingerprint Verification Competition, Proc. 16th Int'l Conf. on Pattern Recognition, Quebec City, Canada, 2002, pp. 811 – 814.
24. W. Y. Choi, D. Moon, K. Y. Moon and Y. Chung, A New Alignment Algorithm of Fuzzy Fingerprint Vault Without Extra Information, Proc. IASTED Int'l Conf. on Artificial Intelligence and Applications (AIA2009), Innsbruck, Austria, 2009.