Fingerprint Image Enhancement:
Segmentation to Thinning

Iwasokun Gabriel Babatunde
Department of Computer Science
Federal University of Technology,
Akure, Nigeria

Akinyokun Oluwole Charles
Department of Computer Science
Federal University of Technology,
Akure, Nigeria

Alese Boniface Kayode
Department of Computer Science
Federal University of Technology,

Olabode Olatubosun
Department of Computer Science
Federal University of Technology,
Akure, Nigeria

Abstract— **Fingerprint has remained a very vital index for human recognition. In the field of security, series of Automatic Fingerprint Identification Systems (AFIS) have been developed. One of the indices for evaluating the contributions of these systems to the enforcement of security is the degree with which they appropriately verify or identify input fingerprints. This degree is generally determined by the quality of the fingerprint images and the efficiency of the algorithm. In this paper, some of the sub-models of an existing mathematical algorithm for the fingerprint image enhancement were modified to obtain new and improved versions. The new versions consist of different mathematical models for fingerprint image segmentation, normalization, ridge orientation estimation, ridge frequency estimation, Gabor filtering, binarization and thinning. The implementation was carried out in an environment characterized by Window Vista Home Basic operating system as platform and Matrix Laboratory (MatLab) as frontend engine. Synthetic images as well as real fingerprints obtained from the FVC2004 fingerprint database DB3 set A were used to test the adequacy of the modified sub-models and the resulting algorithm. The results show that the modified sub-models perform well with significant improvement over the original versions. The results also show the necessity of each level of the enhancement.**

Keyword- AFIS; Pattern recognition; pattern matching; fingerprint; minutiae; image enhancement.

I. Introduction

In the world today, fingerprint is one of the essential variables used for enforcing security and maintaining a reliable identification of any individual. Fingerprints are used as variables of security during voting, examination, operation of bank accounts among others. They are also used for controlling access to highly secured places like offices, equipment rooms, control centers and so on. The result of the survey conducted by the International Biometric Group (IBG) in 2004 on comparative analysis of fingerprint with other biometrics is presented in Fig. 1. The result shows that a substantial margin exists between the uses of fingerprint for identification over other biometrics such as face, hand, iris, voice, signature and middleware [1]. The following reasons had been adduced to the wide use and acceptability of fingerprints for enforcing or controlling security [1]-[4]:

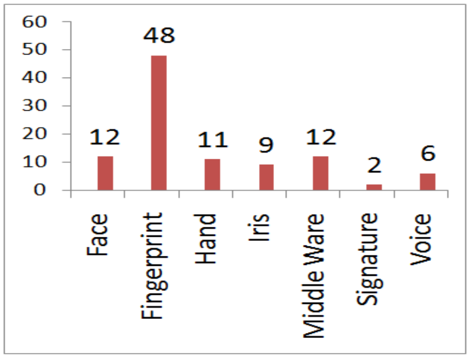


Figure 1: Comparative survey of fingerprint with other biometrics

a) Fingerprints have a wide variation since no twopeople have identical prints.

b) There is high degree of consistency in fingerprints. A person's fingerprints may change in scale but not in relative appearance, which is not the case in other biometrics.

c) Fingerprints are left each time the finger contacts a surface.

d) Availability of small and inexpensive fingerprint capture devices

e) Availability of fast computing hardware

f) Availability of high recognition rate and speed devices that meet the needs of many applications

g) The explosive growth of network and Internet transactions

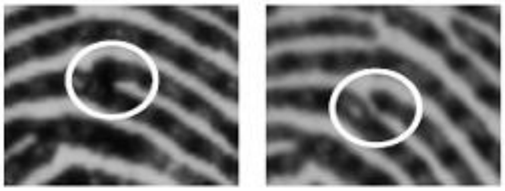
h) The heightened awareness of the need for ease-of-use as an essential component of reliable security.

The main ingredients of any fingerprint used for identification and security control are the features it possesses. The features exhibit uniqueness defined by type, position and orientation from fingerprint to fingerprint and they are classified into global and local features [5]-[7]. Global features are those characteristics of the fingerprint that could be seen with the naked eye. They are the features that are characterized by the attributes that capture the global spatial relationships of a fingerprint. Global features include ridge pattern, type, orientation, spatial frequency, curvature, position and count. Others are type lines, core and delta areas.

The Local Features are also known as Minutia Points. They are the tiny, unique characteristics of fingerprint ridges that are used for positive identification. Local features contain the information that is in a local area only and invariant with respect to global transformation. It is possible for two or more impressions of the same finger to have identical global features but still differ because they have local features (minutia points) that are different. In Fig. 2, ridge patterns (a) and (b) are two different impressions of the same finger (person). A local feature is read as bifurcation in (a) while it appears as a ridge ending in (b).

II. FINGERPRINT IMAGE ENHANCEMENT

Reliable and sound verification of fingerprints in any AFIS is always preceded with a proper detection and extraction of its features. A fingerprint image is firstly enhanced before the features contained in it could be detected or extracted. A well enhanced image will provide a clear separation between the valid and spurious features. Spurious features are those minutiae points that are created due to noise or artifacts and they are not actually part of the fingerprint. This paper adopts with slight modifications, the algorithm implemented in [8]-[9] for fingerprint image enhancement. The overview of the algorithm is shown in Fig. 3. Its main steps include image segmentation, local normalization, filtering and binarization/thinning.



(a) Bifurcation (b) Ridge ending

Figure 2: Different minutiae for different impressions of the same finger

A Image Segmentation

There are two regions that describe any fingerprint image; namely the foreground region and the background region. The foreground regions are the regions containing the ridges and valleys. As shown in Fig. 4, the ridges are the raised and dark regions of a fingerprint image while the valleys are the low and white regions between the ridges. The foreground regions often referred to as the Region of Interest (RoI) is shown for the image presented in Fig. 5. The background regions are mostly the outside regions where the noises introduced into the image during enrolment are mostly found. The essence of segmentation is to reduce the burden associated with image enhancement by ensuring that focus is only on the foreground regions while the background regions are ignored.

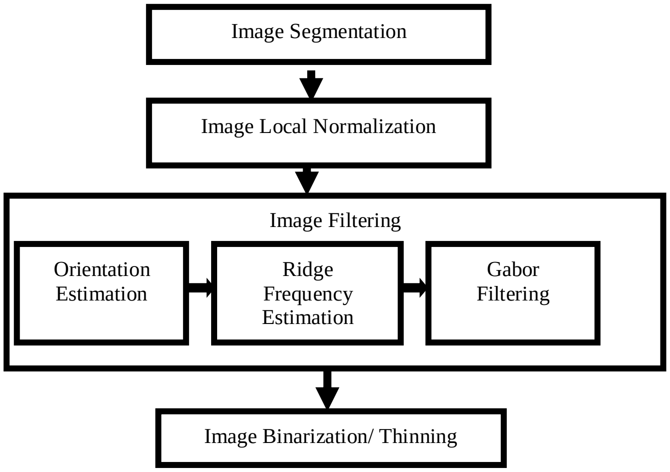


Image Segmentation
Image Local Normalization
Image Filtering
Orientation Estimation
Ridge Frequency Estimation
Gabor Filtering
Image Binarization/ Thinning

Figure 3: The conceptual diagram of the fingerprint enhancement algorithm

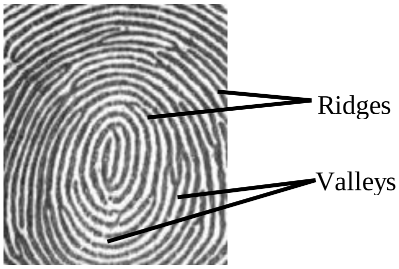


Figure 4: Ridges and valleys on a fingerprint image

The background regions possess very low grey-level variance values while the foreground regions possess very high grey-level variance values. A block processing approach used in [8]-[9] is adopted in this research for obtaining the grey-level variance values. The approach firstly divides the image into blocks of size W x W and then the variance V(k) for each of the pixels in block k is obtained from:

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

$$M(k) = \frac{1}{W^2}\sum_{a=1}^{W}\sum_{b=1}^{W}J(a,b) \tag{2}$$

I(i,j) and J(a,b) are the grey-level value for pixel i,j and (a,b) respectively in block k.

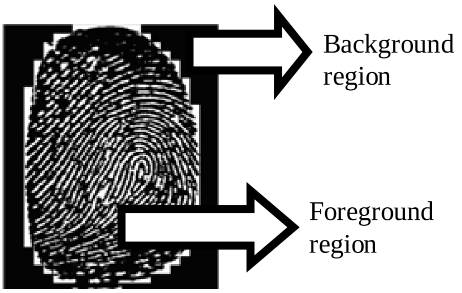


Figure 5: A fingerprint image and its foreground and background regions

A. Image Local Normaliuzation

Normalization is performed on the segmented fingerprint image ridge structure so as to standardize the level of variations in the image grey-level values. By normalization, the grey-level values are made to fall within certain range that is good enough for improved image contrast and brightness. The first of the tasks of image normalization implemented in [8]-[9] and adopted for this research is the division of the segmented image into blocks of size S x S. The grey-level value for each pixel is then compared with the average grey-level value for the host block. For a pixel I(i,j) belonging to a block of average grey-level value of M, the result of comparison produced a normalized grey-level value N(i,j) defined by the formula:

$$N(i,j) = \begin{cases} M_0 + \sqrt{\dfrac{V_0(I(i,j)-M)^2}{V}} & if\ I(i,j) > M \\ M_0 - \sqrt{\dfrac{V_0(I(i,j)-M)^2}{V}} & otherwise \end{cases} \tag{3}$$

where an assumed value of M_0 is set for the desired mean and an assumed value of V_0 is set for the desired variance.

B. Image Filtering

Normalized fingerprint image is filtered for enhancement through removal of noise and other spurious features. Filtering is also used for preserving the true ridge and valley structures. The fingerprint image filtering structure adopted for this research is in the following phases:

1) Orientation Estimation: Orientation estimation is the first of the prerequisites for fingerprint image filtering. In every image, the ridges form patterns that flow in different directions. The orientation of a ridge at location x,y is the direction of its flow over a range of pixels as shown in Fig. 6.

The Least Square Mean (LSM) fingerprint ridge orientation estimation algorithm proposed and implemented in [8]-[9] was slightly modified and used in this research. The modified algorithm involves the following steps:

a) Firstly, blocks of size S x S were formed on the normalized fingerprint image.

b) For each pixel, (p,q) in each block, the gradients ∂_x(p,q) and ∂_y(p,q) were computed as the gradient magnitudes in the x and y directions, respectively.

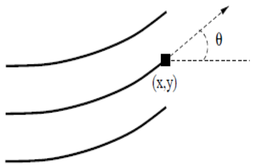


Figure 6: The orientation of a ridge pixel in a fingerprint

∂_x(p,q) was computed using the horizontal Sobel operator while ∂_y(p,q) was computed using the vertical Sobel operator.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Horizontal Sobel Operator Vertical Sobel Operator

c) The local orientation of a pixel in a fingerprint image was computed by using its S x S neighborhood in [8]-[9]. This was slightly modified in this research by dividing the image into S x S blocks and the local orientation for each block centered at pixel I(i,j) was then computed from:

$$V_x(i,j) = \sum_{p=i-\frac{S}{2}}^{i+\frac{S}{2}} \sum_{q=j-\frac{S}{2}}^{j+\frac{S}{2}} 2\partial_x(p,q)\partial_y(p,q) \tag{4}$$

$$V_y(i,j) = \sum_{p=i-\frac{S}{2}}^{i+\frac{S}{2}} \sum_{q=j-\frac{S}{2}}^{j+\frac{S}{2}} \partial_x^2(p,) - \partial_y^2(p,q) \tag{5}$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\frac{V_y(i,j)}{V_x(i,j)} \tag{6}$$

where $\Theta(i,\ j)$ is the least square estimate of the local orientation at the block centered at pixel (i, j).

d) The orientation image is then converted into a continuous vector field defined by:

$$\varphi_x(i,j) = \cos\big(2\theta(i,j)\big), \tag{7}$$

$$\varphi_y(i,j) = \sin\big(2\theta(i,j)\big), \tag{8}$$

where φ_x and φ_y are the x and y components of

the vector field, respectively.

e) Gaussian smoothing is then performed on the vector field as follows:

$$\varphi_X'(i,j) = \sum_{p=-\frac{S_\varphi}{2}}^{\frac{S_\varphi}{2}} \sum_{q=-\frac{S_\varphi}{2}}^{\frac{S_\varphi}{2}} G(p,q)\varphi_x(i-ps, j-qs),. \tag{9}$$

$$\varphi_y^{'}(i,j) = \sum_{p=-\frac{S_\varphi}{2}}^{\frac{S_\varphi}{2}} \sum_{q=-\frac{S_\varphi}{2}}^{\frac{S_\varphi}{2}} G(p,q)\varphi_y(i-ps, j-qs), \qquad (10)$$

where *G* is a Gaussian low-pass filter of size $S_\varphi \, X \, S_\varphi$.

f) The orientation field O of the block centered at pixel (i,j) is finally smoothed using the equation:

$$O(i,j) = \frac{1}{2}\tan^{-1}\frac{\varphi_y'(i,j)}{\varphi_x'(i,j)} \qquad (11)$$

2) Ridge Frequency Estimation: The second prerequisite for fingerprint image filtering is the ridge frequency estimation. In any fingerprint image, there is a local frequency of the ridges that collectively form the ridge frequency image. The ridge frequency is obtained from the extraction of the ridge map from the image. The extraction of the ridge map involves the following steps:

a) Compute the consistency level of the orientation field obtained from the first prerequisite in the local neighborhood of a pixel (p,q) with the following formula:

$$C_o(p,q) = \frac{1}{n^2}\sqrt{\sum_{(i,j)\in W} |\theta(i,j)-\theta(p,q)|^2} \qquad (12)$$

$$|\theta(i,j) - \theta(p,q)| = \begin{cases} d & if\ d < 180 \\ d - 180 & otherwise \end{cases} \qquad (13)$$

$$d = (\theta(i,j) - \theta(p,q) + 360)\ mod\ 360 \qquad (14)$$

where W represents the local neighborhood around (p,q), which is an n x n local window, θ(i,j) and θ(p,q) are local ridge orientations at pixels (i,j) and (p,q) respectively.

b) If the consistency level is below a certain threshold F_c, then the local orientations in this region are re-estimated at a lower image resolution level until the consistency is above F_c. After the orientation field is obtained, the following two adaptive filters are applied to the image:

$$h_t(p,q,i,j) = \begin{cases} \frac{-1}{\sqrt{2\pi\delta}} e^{\frac{-1}{\delta^2}}, & if\ i = l(j) - d,\ j \in \rho \\ \frac{1}{\sqrt{2\pi\delta}} e^{\frac{-1}{\delta^2}}, & if\ i = l(j),\ j \in \rho \\ 0, & otherwise, \end{cases} \qquad (15)$$

$$h_b(p,q,i,j) = \begin{cases} \frac{-1}{\sqrt{2\pi\delta}} e^{\frac{-1}{\delta^2}} & if\ i = l(j) + d,\ j \in \rho \\ \frac{1}{\sqrt{2\pi\delta}} e^{\frac{-1}{\delta^2}}, & if\ i = l(j),\ j \in \rho \\ 0, & otherwise, \end{cases} \qquad (16)$$

$$l(j) = j\ tan(\theta(p,q)); \qquad (17)$$

$$d = \frac{Y}{2\cos(\theta(p,q))}, \qquad (18)$$

$$\rho = Y\left[\left|\frac{\sin(\theta(p,q)}{-2}\right|, \left|\frac{\sin(\theta(p,q)}{2}\right|\right] \qquad (19)$$

The two filters are capable of stressing under different condition the local maximum grey level values along the normal direction of the local ridge orientation. The normalized image is first convolved with these two masks, h_t (p,q, i, j) and h_b (p,q, i, j). If both the grey level values at pixel (p,q) of the convolved images are larger than a certain threshold F_{ridge}, then pixel (p,q) is labeled as a ridge.

3) Orientation Estimation: Having obtained the prerequisites, Gabor filtering is then used to improve or enhance the fingerprint image to a finer structure. It involves the removal of noise and artifacts. The general form of Gabor filter is:

$$G(x,y\colon f,\theta) = exp\left\{\frac{1}{2}\left[\frac{a^2}{\delta_x^2} + \frac{b^2}{\delta_y^2}\right]\right\}cos(2\pi fa) \qquad (20)$$

where f is the frequency of the cosine wave along the direction θ from the *x*-axis, and δ_x and δ_y are the space constants along *x* and *y* axes respectively. a= xsinΘ + ycosΘ and b= xcosΘ + ysinΘ.

The values of the space constants δ_x and δ_y for the Gabor filters were empirically determined as each is set to about half the average inter-ridge distance in their respective direction. δ_x and δ_y are obtained from $\delta_x = k_x F$ and $\delta_y = k_y F$ respectively. F is the ridge frequency estimate of the original image, and k_x *and* k_y are constant variables. The value of δ_x determines the degree of contrast enhancement between ridges and valleys while the value of δ_y determines the amount of smoothing applied to the ridges along the local orientation.

C. Image Binarization/Thinning

The image obtained from the Gabor filtering stage is binarized and thinned to make it more suitable for feature extraction. The method of image binarization proposed in [10] is employed. The Method sets the threshold (T) for making each cluster in the image as tight as possible, thereby minimizing their overlap. To determine the actual value of T, the following operations are performed on set of presumed threshold values:

a) The pixels are separated into two clusters according to the threshold.

b) The mean of each cluster are determined.

c) The difference between the means is squared.

d) The product of the number of pixels in one cluster and the number in the other is determined.

The success of these operations depends on the difference between the means of the clusters. The optimal threshold is the one that maximizes the between-class variance or, conversely, the one that minimizes the within-class variance. The *within-class* variance of each of the cluster is then calculated as the weighted sum of the variances from:

$$\sigma^2_{within}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T) \qquad (21)$$

$$n_B(T) = \sum_{i=0}^{T-1} p(i) \qquad (22)$$

$$n_O(T) = \sum_{i=T}^{N-1} p(i) \qquad (23)$$

$$\sigma^2_B(T) = the\ variance\ of\ the\ pixels\ in\ the\ background\ (below) threshold$$

$$\sigma^2_O(T) = the\ variance\ of\ the\ pixels\ in\ the\ foreground\ (above) threshold$$

p(i) is the pixel value at location i, N is the intensity level and $[0, N-1]$ is the range of intensity levels. The between-class variance, which is the difference between the within-class variance and the total variance of the combined distribution, is then obtained from:

$$\sigma^2_{between}(T) = \sigma^2 - \sigma^2_{within}(T) \tag{24}$$

$$= n_B(T)[A] + n_O(T)[B] \tag{25}$$

$$A = (\mu_B(T) - \mu)^2 \tag{26}$$

$$B = (\mu_O(T) - \mu)^2 \tag{27}$$

where σ^2 is the combined variance, μ_B(T) is the combine mean for cluster T in the background threshold, μ_O(T) is the combine mean for cluster T in the foreground threshold and μ is the combined mean for the two thresholds. The between-class variance is simply the weighted variance of the cluster means themselves around the overall mean. Substituting $\mu = n_B(T)\mu_B(T) + n_O(T)\mu_O(T)$ into Equation 25, the result is:

$$\sigma^2_{between}(T) = n_B(T)n_O(T)[\mu_B(T) - \mu_O(T)]^2 \tag{28}$$

Using the following simple recurrence relations, the between-class variance is successfully updated by manipulating each threshold T using a constant value p as follows:

$$n_B(T+1) = n_B(T) + p \tag{29}$$

$$n_O(T+1) = n_O(T) - p \tag{30}$$

$$\mu_B(T+1) = \frac{\mu_B(T)n_B(T) + pT}{n_B(T+1)} \tag{31}$$

$$\mu_O(T+1) = \frac{\mu_O(T)n_O(T) - pT}{n_O(T+1)} \tag{32}$$

III. Experimental Results

A slightly modified version of the fingerprint enhancement algorithm used in [8]-[9] was implemented in this research by using MATLAB Version 7.2 on the Windows Vista Home Basic operating system. The experiments were performed on a Pentium 4 – 1.87 GHz processor with 1024MB of RAM. The purpose of the fingerprint enhancement experiments is to analyze the performance of the modified algorithm under different conditions of images as well as generate the metrics that could serve the basis for the comparison of the results from the research with results from related works. Two sets of experiment were conducted for the performance analysis. The first set of experiments was on synthetic images. The orientation estimation, ridge frequency estimation and Gabor filtering experiments all employed the *circsine* function [11] to generate the synthetic images. The major arguments passed into the *circsine* function include a number for the size of the square image to be produced, a number for the wavelength in pixels of the sine wave and an optional number specifying the standard pattern of behaviour to use in calculating the radius from the centre. This defaults to 2, resulting in a circular pattern while large values give a square pattern. Where necessary, the MATLAB *imnoise* function was also used to generate noise and artifacts on synthetic images. The arguments passed into the *imnoise* function include the image on which noise is to be generated, noise type and noise level. The second set of experiments was on the FVC2004 fingerprint database DB3 set A.

Fig. 7(a), 7(b) and 7(c) are synthetic images of size 200 x 200 and wavelength 8. They were obtained with the *imnoise* function using the salt and pepper noise level of 0, 0.2 and 0.4 respectively. The results of the ridge orientation estimation experiments on each of the three images are shown in Fig. 7(d), 7(e) and 7(f) respectively. These results show that for noise levels of 0 and 0.20, the modified ridge orientation algorithm effectively generated ridge orientation estimates that are very close to the actual orientations. However, for image with noise level of 0.4, the result shows a substantial number of ridge orientation estimates that significantly differ from the actual orientations. These results show that the performance of the algorithm depends on the image noise level. When the noise level on the image is within reasonable range, the algorithm does well while it fails when the noise level rises beyond the threshold which was found to be 0.29.

The efficiency of the ridge orientation estimation algorithm was quantitatively measured by estimating the Mean Square Error (MSE) which represents the difference between the estimated and actual ridge orientation values in radians. Mean square error estimation results for the synthetic image shown in Fig. 7(a) under different conditions of noise for both the pixel processing approach in [8] and the block processing approach formulated for this research are shown in Table 1. The increasing mean square errors in both cases indicate that the accuracy of the two approaches decreases with increase in the noise level. It is also revealed that the orientation estimate is closer to the actual value for the block processing approach at any noise level. With lower standard deviation of 0.0393 for its MSE values, it is established that the block processing approach performs better than the pixel processing approach with MSE values of standard deviation of 0.1686. This higher performance is attributed to the fact that in the block processing approach, the degree of variation in the orientation estimates for pixels in a block is reduced to zero as each pixel assumes the orientation estimate for the center pixel of its host block. This significantly increases the ability of the algorithm to estimate the ridge orientation close to the actual value.

Fig. 7(g), 7(h) and 7(i) present the results of the ridge frequency estimation experiments on the synthetic images of different noise levels shown in Fig. 7(a), 7(b) and 7(c) respectively. In the ridge frequency estimates shown in Fig. 7(g) and Fig. 7(h), there is uniformity between most of the estimated frequency values for each 32 *x* 32 block as against Fig. 7(i) which shows irregular patterns. Fig. 7(g), 7(h) and 7(i) produced MSE value of 0.0006, 0.0017 and 0.0077 respectively. The ridge frequency estimates and the increasing MSE values reveal that the performance of the ridge frequency estimation algorithm decreases with increase in the image noise level. Generally, below the noise level of 0.30, it was discovered that the algorithm produced relatively uniform ridge frequency estimates as shown in Fig. 7(g) and 7(h).

When the noise level equals or exceeds 0.3, the orientation estimation algorithm produced non uniform results as shown in Fig. 7(i). Since the performance of the ridge frequency estimation algorithm depends significantly on the performance

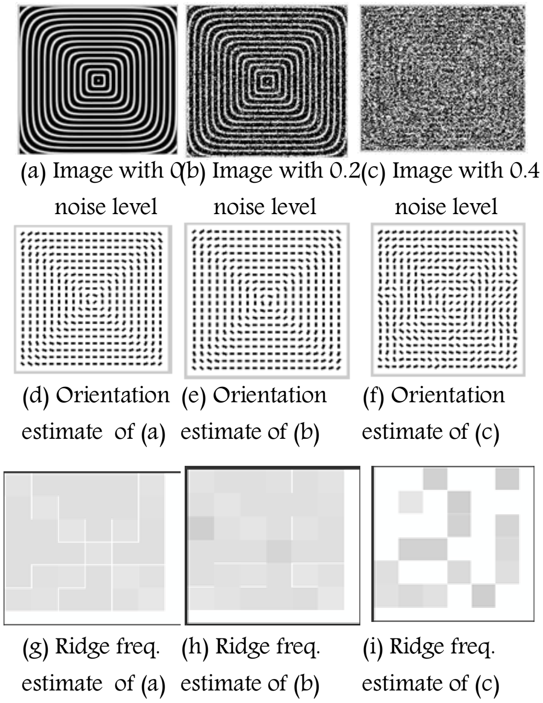

(a) Image with 0 noise level
(b) Image with 0.2 noise level
(c) Image with 0.4 noise level
(d) Orientation estimate of (a)
(e) Orientation estimate of (b)
(f) Orientation estimate of (c)
(g) Ridge freq. estimate of (a)
(h) Ridge freq. estimate of (b)
(i) Ridge freq. estimate of (c)

Figure 7: Orientation and ridge frequency estimates for synthetic images of different noise levels

of the ridge orientation estimation algorithm, the failure of the orientation estimation algorithm explains the failure of the ridge frequency estimation algorithm for noise level above 0.29.

The MSE results of the implementation of the ridge frequency estimation algorithm on the synthetic image shown in Fig. 7(a) under different conditions of noise in [8] and the current study are shown in Table 2. It is revealed that there is increase in the MSE values as the noise level increases for the two cases. This translates to diminishing performance due to increasing margin between the actual and the estimated ridge frequency values. It is also shown that the MSE value is significantly lower in all cases for the current study than in [8]. With lower standard deviation of 0.0032 for its MSE values, it is established that the implementation of the ridge frequency estimation algorithm in the current study yields better results than its implementation in [8] which yielded a standard deviation of 2.8514 for its MSE values. This improvement is attributed to the superior performance of the block processing approach to ridge orientation estimation over the pixel processing approach.

The performance of the Gabor filtering algorithm on a zero, medium and high quality synthetic image of size 410 x 410 and wavelength 15 is presented in Fig. 8(d), 8(e) and 8(f) respectively. Parameter values of k_x =0.45 and k_y =0.45 were used to obtain these results. The results presented in Fig. 8(d) and 8(e) reveal that with zero or medium level noise density, the filter effectively removed the noise from the image and enhanced it to a level that is comparable with the original image. This effectiveness is partly due to the previous accurate estimation of the ridge orientation and the ridge frequency for zero or medium noise level images.

However, the experimental result presented in Fig. 8(f) reveals that when the filter is applied to images with higher noise level, the filter is unable to remove the noise effectively as it produced a significant amount of spurious features. This ineffectiveness is due to the inaccurate estimation of the ridge orientation and the ridge frequency at higher noise levels. When experiments were performed on real fingerprint images, like in [8], the best results were obtained for image segmentation with variance threshold of 100. This threshold value provided the best segmentation results in terms of differentiating between the foreground and the background regions as shown in Fig. 9(b). The results of segmentation using threshold value of 95 and 105 are presented in Fig. 9 (c) and 9(d) respectively.

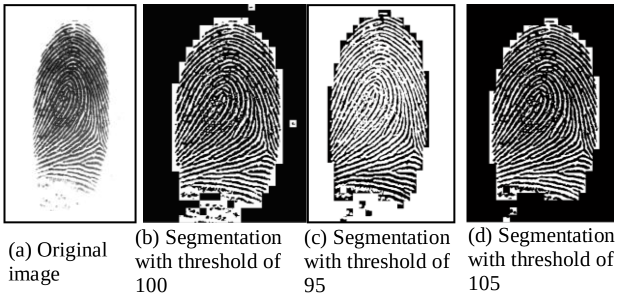

(a) Original image
(b) Segmentation with threshold of 100
(c) Segmentation with threshold of 95
(d) Segmentation with threshold of 105

Fig. 9: Results of segmentation with different threshold

TABLE 1: COMPARISON OF MEAN SQUARE ERROR FOR RIDGE ORIENTATION ESTIMATES

Noise level	Mean Square Error	
	Raymond [8]	Current study
0.00	0.0003	0.0000
0.05	0.0009	0.0006
0.10	0.0032	0.0008
0.15	0.0102	0.0019
0.20	0.0246	0.0015
0.25	0.0691	0.0026
0.30	0.1722	0.0405
0.35	0.2330	0.0521
0.40	0.3041	0.0661
0.45	0.4124	0.0803
0.50	0.4262	0.1085

TABLE 2: COMPARISON OF MEAN SQUARE ERROR FOR RIDGE FREQUENCY ESTIMATES

Noise level	Mean Square Error	
	Raymond[8]	Current Study
0.00	0.0100	0.0006
0.05	0.0211	0.0009
0.10	0.0465	0.0011
0.15	0.0820	0.0012
0.20	0.1702	0.0017
0.25	1.1149	0.0033
0.30	2.0229	0.0041
0.35	4.1149	0.0060
0.40	5.8543	0.0077
0.45	6.1098	0.0080
0.50	7.2616	0.0090

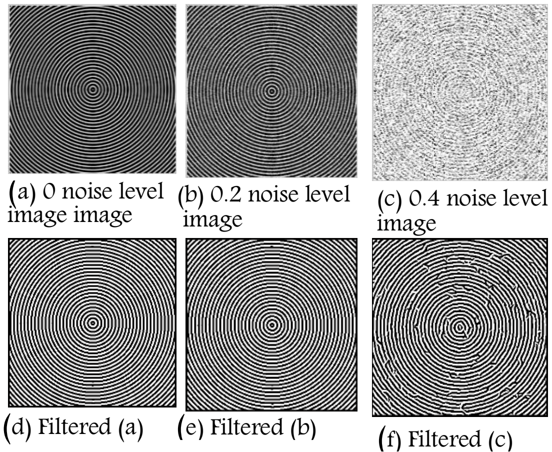

(a) 0 noise level image image
(b) 0.2 noise level image
(c) 0.4 noise level image
(d) Filtered (a)
(e) Filtered (b)
(f) Filtered (c)

Figure 8: Results of applying a Gabor filter on synthetic images of different noise levels.

These results show inappropriate segmentation due to inaccurate variance thresholds. With lower threshold of 95, some of the background regions have been segmented to the foreground while some foreground regions are also segmented to the background under higher threshold of 105.

The result of the normalization experiment on the fingerprint images shown in Fig. 9(a) is presented in Fig. 10(a). The desire mean of zero and variance of one used in [8] were adopted and used to normalize the ridges in the images. During normalization, all positions are evenly shifted along the horizontal axis, which makes the structure of the ridges and valleys to become well and suitably positioned.

The histogram plots of the original and the normalized images are shown in Fig. 10(b) and 10(c) respectively. The histogram plot of the original image shows that all the intensity values of the ridges show irregular frequency values and also fall within the right hand side of the 0–255 scale, with no pixels in the left hand side. This leads to an image with a very low contrast.

The histogram plot of the normalized image shows that the range of intensity values for the ridges has been adjusted between 0-1 scale such that there is a more evenly and balanced distribution between the dark and light pixels and that the ridge frequencies fall within close values. The normalized image histogram plots also show that the normalization process does distribute evenly the shape of the original image. The positions of the values are evenly shifted along the *x*-axis, which means the structure of the ridges and valleys are now well and suitably positioned. This shifted and improved positioning lead to images with a very high contrast shown in Fig. 10(a).

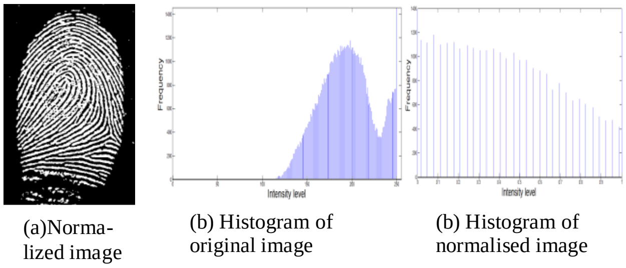


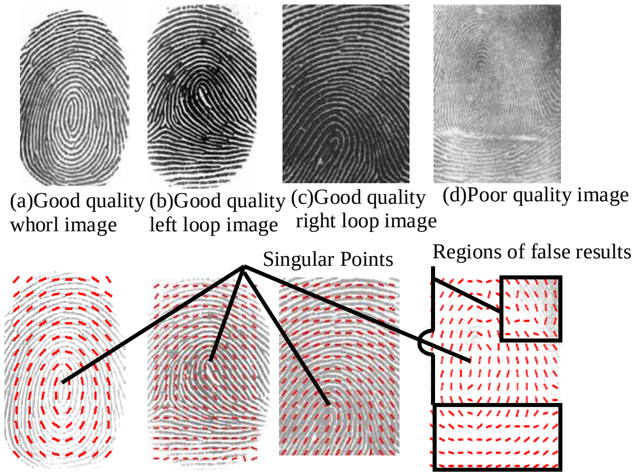
Figure. 10: Normalized image and the histogram plots

The orientation fields for the real fingerprint images were obtained around their singular points since they are prominent features used by any AFIS for fingerprint classification and matching. Good quality images are shown in Fig. 11(a), 11(b) and 11(c). Their orientation estimates are shown in Fig. 11(e), 11(f) and 11(g) respectively.

At the singular points, the orientation field is discontinuous and unlike the normal ridge flow pattern, the ridge orientation varies significantly. From these results, it is observed that there exists no deviation between the actual fingerprint ridge orientation and the estimated orientation of the vectors. In both cases, the algorithm produces accurate estimates of the orientation vectors such that they flow smoothly and consistently with the direction of the ridge structures in the images. In the superimposed version of images in Fig. 11(e), 11(f), 11(g) and 11(h), the contrast of the original image is lowered in each case. This was done to improve the visibility of the orientation vectors against the background.

The ridge orientation estimate for poor quality image shown in Fig. 11(d) is presented in Fig. 11(h). The estimate indicates a fairly smooth orientation field in some well-defined regions while it gives misleading results in areas of very poor quality as evident in the top-left and bottom regions of the estimate. The orientation estimates resulting from the pixel processing approach in [8] and the block processing approach of the current study for the image shown in Fig. 12(a) are presented in Fig. 12(b) and 12(c) respectively.

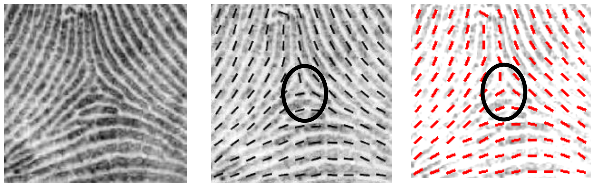
Visual inspection of these results reveals that the two methods did well in the ridge orientation estimation. However, the orientation is observed to be closer to the actual orientation in block processing than pixel processing in some regions especially the core areas represented by the inserted circles. The reason adduced to this is that assigning equal orientation estimate for pixels in a block rather than maintaining different values is better and able to take the estimates closer to their actual values.



(a)Good quality whorl image (b)Good quality left loop image (c)Good quality right loop image (d)Poor quality image

(e) Orientation estimate of (a) (f) Orientation estimate of (b) (g) Orientation estimate of (c) (h) Orientation estimate of (d)

Figure 11: Fingerprint images and their orientation estimates



(a) Original Image

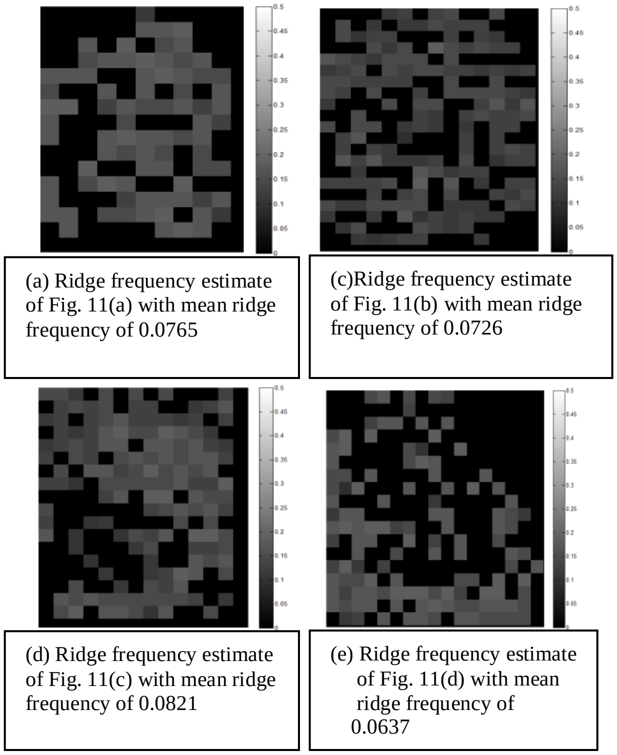
(b) Orientation estimate obtained in [8]

(c) Orientation estimate for current study

Fig. 12: Orientation estimates for pixel and block processing approaches

Visual inspection of the results for the ridge frequency estimates for real fingerprint images shown in Fig. 11(a), 11(b), 11(c) and 11(d) are shown in Fig. 13(a), 13(b), 13(c) and 13(d) respectively. The mean ridge frequency (MRF), which is the average of the image ridge frequencies, is also presented for each image. It is noted that the MRF values differ for all the images. This difference is attributed to the fact that fingerprints exhibit variation in their average ridge frequency characteristics and contrast levels. The intensities of frequency differ for blocks or regions within same image. Some blocks or regions exhibit high contrast while others exhibit low contrast. Based on these, the synthetic images are more appropriate for the evaluation of the accuracy of the ridge frequency estimation algorithm.

Fig. 14 reveals the extent to which the filtering algorithm was able to remove noise from the images shown in Fig. 11 for different values of k_x and k_y. The results shown in Fig. 14(a), 14(b), 14(c) and 14(d) were obtained using parameter values of k_x = 0.45 and k_y = 0.45. With these values, it is shown that the contrast level between ridges and valleys for each of the images is neither too high nor too low. Appropriate level of smoothing is also applied to the ridges along the local orientation.



(a) Ridge frequency estimate of Fig. 11(a) with mean ridge frequency of 0.0765

(c)Ridge frequency estimate of Fig. 11(b) with mean ridge frequency of 0.0726

(d) Ridge frequency estimate of Fig. 11(c) with mean ridge frequency of 0.0821

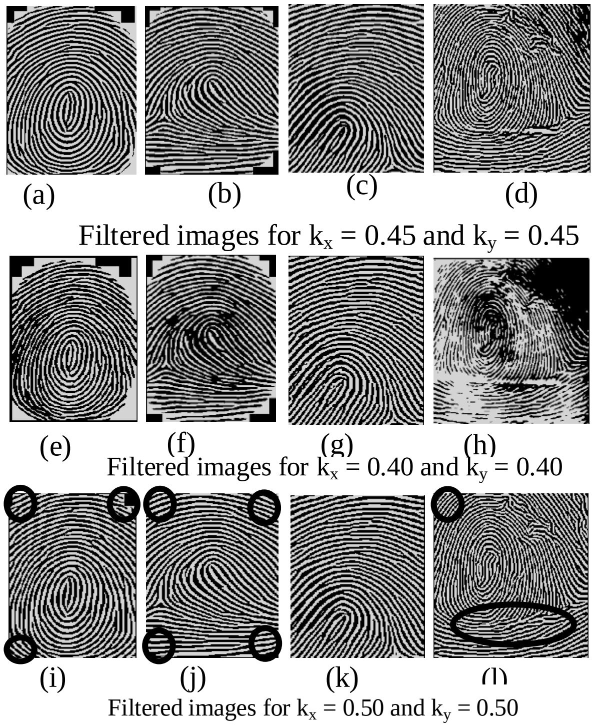
(e) Ridge frequency estimate of Fig. 11(d) with mean ridge frequency of 0.0637

Fig. 13: Ridge frequency estimate of selected images

With lower parameter values of k_x = 0.40 and k_y = 0.40, it is shown in Fig. 14(e), 14(f), 14(g) and 14(h) that the contrast level between ridges and valleys is too low and this explains why there is a number of dark regions. The degree of smoothing is also poor as there is a good number of overlapping ridges in the filtered images. When k_x=0.5 and k_y=0.5 were used, the results of the filtering experiments shown in Fig. 14(i), 14(j), 14(k) and 14(l) reveal that there is no significant difference from the results obtained for k_x=0.45 and k_y=0.45 except that some regions described by circles appear to be excessively smoothened.

It is therefore stated that based on the modified algorithm, parameter values of k_x=0.45 and k_y=0.45 are most appropriate for image filtering as against k_x=0.50 and k_y=0.50 proposed in [8]-[9]. This reduction in parameter values is due to the better performance of the block processing approach in the orientation and ridge frequency estimations as attested to by the results in Table 1 and Table 2. Results presented in Fig. 14 show that the filtering algorithm is able to smoothen to a fine level with appropriate parameter values for good quality fingerprint images.

When the quality degrades like the one shown in Fig. 11(d), the performance of the algorithm diminishes as it produces images with inappropriately filtered regions as shown in Fig. 14(d). This is also corroborated with reasons adduced to the values presented in Table 1 and Table 2.



Filtered images for $k_x = 0.45$ and $k_y = 0.45$

Filtered images for $k_x = 0.40$ and $k_y = 0.40$

Filtered images for $k_x = 0.50$ and $k_y = 0.50$

Fig. 14: Filtered images with different values for k_x and k_y

The results of the binarization experiments for the images shown in Fig. 11 are presented in Fig. 15. Visual inspection of the results shows that the binarization algorithm perfectly separated the ridges (black pixels) from the valleys (white pixels).

To obtain these results, the grey-level value of each pixel in the filtered image is examined and, if the value is greater than the threshold value 1, then the pixel value is set to a binary value one; otherwise, it is set to zero. The threshold value successfully made each cluster as tight as possible and also eliminated all overlaps. The threshold value of 1 was taken after a careful selection from a series of within and between class variance values ranging from 0 to 1 that optimally supported the maximum separation of the ridges from the valleys. The clear separation of the ridges from the valleys verifies the correctness of the algorithm as proposed in [9] and implemented in this research.

The results of the thinning experiment on each of the images shown in Fig. 11 are presented in Fig. 16(a-d). The MATLAB's *bwmorph* operation using the 'thin' option was used to generate the thinned images. These results show that the ridge thickness in each of the images has been reduced to its smallest form or skeleton (one pixel wide). It is also shown that the connectivity of the ridge structures is well preserved. Fig. 16(e-h) shows the results of performing binarization experiments on the raw images without the enhancement stages. In contrast to Fig. 15(a-d), the binary images in Fig. 16(e-h) are not well connected and contain significant amount of noise and corrupted elements. Consequently, when thinning is applied to these binary images, the results in Fig. 16(i-l) show that the accurate extraction of minutiae would not be possible due to the large number of spurious features produced. Thus, it is shown that employing the image enhancement stages prior to thinning is effective for accurate and speedy extraction of minutiae. Consequently, when thinning is applied to these binary images, the results in Fig. 16(i-l) show that the accurate extraction of minutiae would not be possible due to the large number of spurious features produced.

Thus, it is shown that employing the image enhancement stages prior to thinning is effective for accurate and speedy extraction of minutiae.

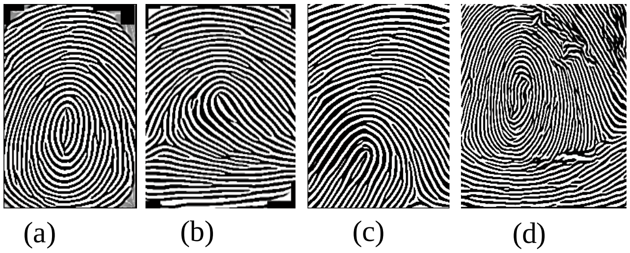
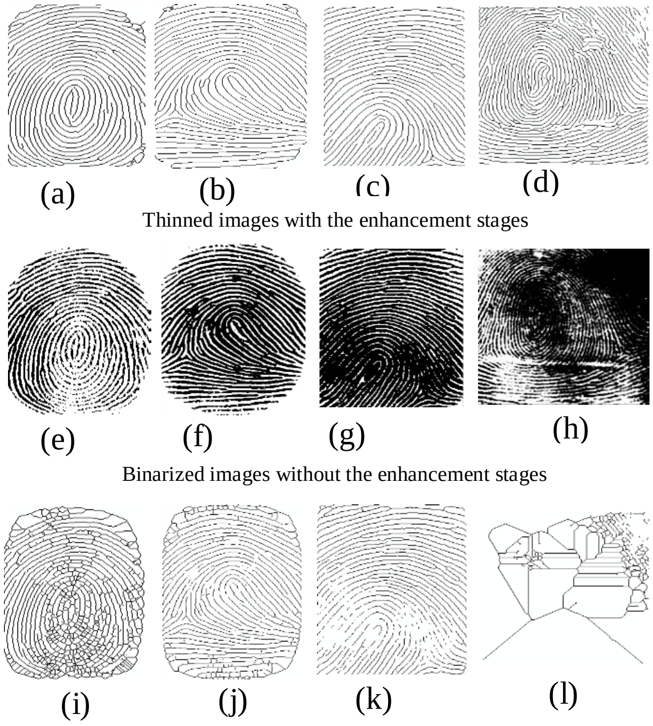


Fig. 15: Results of Binarization for images shown in Fig. 11(a), 11(b), 11(c) and 11(d)



Thinned images with the enhancement stages

Binarized images without the enhancement stages

Thinned images without the enhancement stages

Fig. 16: Thinned and Binarized images

IV. CONCLUSION

This paper discusses the results of the modification and verification of the fingerprint enhancement algorithm developed and implemented in [8]-[9]. Some stages of the algorithm were slightly modified for improved performance. For instance, block processing approach was introduced into the orientation estimation algorithm in place of the pixel processing approach.

While the pixel processing approach subjects each pixel in the image to orientation estimation, the block processing approach firstly divides the image into S x S blocks and obtains the orientation estimate for the center pixel. This resulted in higher performance as attested to by the Tables of MSE for ridge orientation and frequency estimates. Parameter values of k_x =0.45 and k_y =0.45 were found to perform well in the image filtering experiments as against k_x =0.5 and k_y =0.5.

The results of the experiments conducted for image segmentation, normalization, ridge orientation estimation, ridge frequency estimation, Gabor filtering, binarization and thinning on synthetic and real fingerprint images reveal that with free or minimal noise level, the algorithms perform well. Improved performance is specifically noticed for the modified ridge orientation estimation algorithm. It is also established that each stage of the enhancement process is important for obtaining a perfectly enhanced image that is acceptable and presentable to the features extraction stage. The results obtained from the final stage of thinning show that the connectivity of the image ridge structure has been preserved and improved at each stage.

REFERENCES

[1] C. Roberts: ‘Biometrics’: (http://www.ccip.govt.nz/newsroom/informoationnotes/2005/biometrics.pdf, Accessed: July, 2009

[2] C. Michael and E. Imwinkelried: ‘Defence practice tips, a cautionary note about fingerprint analysis and reliance on digital technology’, Public Defense Backup Centre Report., 2006

[3] M. J. Palmiotto: ‘Criminal Investigation’. Chicago: Nelson Hall, 1994

[4] Salter D.: ‘Fingerprint – An Emerging Technology’, Engineering Technology, New Mexico State University, 2006

[5] J. Tsai-Yang, and V. Govindaraju: A minutia-based partial fingerprint recognition system, Center for Unified Biometrics and Sensors, University at Buffalo, State University of New York, Amherst, NY USA 14228, 2006

[6] O. C Akinyokun, C. O. Angaye and G. B. Iwasokun: ‘A Framework for Fingerprint Forensic’; Proceeding of the First International Conference on Software Engineering and Intelligent System, organized and sponsored by School of Science and Technology, Covenant University, Ota, Nigeria, 2010, pages 183-200.

[7] O. C. Akinyokun and E. O. Adegbeyeni: Scientific Evaluation of the Process of Scanning and Forensic Analysis of Fingerprints on Ballot Papers’, Proceedings of Allied Academies International Conference on Legal, Ethics and Regulatory Measures in New Orleans, April 8 - 10, 2009; Vol. 13; No. 1; pages 1 - 13.

[8] L. Hong, Y. Wan and A. Jain: ‘Fingerprint image enhancement: Algorithm and performance evaluation’; Pattern Recognition and Image Processing Laboratory, Department of Computer Science, Michigan State University, 2006, pp1-30

[9] R. Thai: Fingerprint Image Enhancement and Minutiae Extraction, PhD Thesis Submitted to School of Computer Science and Software Engineering, 2003, University of Western Australia.

[10] X. Xu : ‘Image Binarization using Otsu Method’. Proceedings of NLPR-PAL Group CASIA Conference, 2009, pp345-349

[11] P. Kovesi: ‘MATLAB functions for computer vision and image analysis’, School of Computer Science and Software Engineering, University of Western Australia, http:/www.cs.uwa.edu.au/~pk/Research/MatlabFns/Index.html, Accessed: 20 February 2010.