Development of an Automated Fingerprint Verification System

Martin Saveski

13 May 2010

Table of Contents

1. Introduction 3

2. Fingerprint Representation 4

3. Image Segmentation 5

4. Image Enhancement 7

   4.1 Normalization 8

   4.2 Orientation Estimation 9

   4.3 Ridge Frequency Estimation 11

   4.4 Filtering 12

   4.5 Binarization 13

   4.6 Skeletonization 14

5. Minutiae Extraction 15

6. Fingerprint Matching 17

7. Evaluation of the AFVS Developed 19

8. Conclusions and Future Work 21

9. References 22

1. Introduction

Biometric recognition, or simply *biometrics*, refers to the use of distinctive anatomical and behavioral characteristics or identifiers for automatically recognizing a person (Maltoni et al., 2009). Due to the impossibility of misplacing, falsifying, or sharing biometric identifiers they are considered more reliable for person recognition than traditional token or knowledge based methods. One of the biometric technologies which have been successfully developed and deployed is the fingerprint recognition. Fingerprints are considered to be immutable i.e. persistent over time and individual, having unique ridge details. As reported in (Leung et al., 1991) the probability that two fingerprints are identical is 1 in 1.9×10^{15}. The labor intensive and slow process of manual recognition of fingerprints has inspired a lot of research on building *Automated Fingerprint Recognition Systems* (AFRS) in the past four decades. A major contribution in the progress in this field has the *International Competition for Fingerprint Verification Algorithms* (FVC) which established common benchmark and allowed researches to compare their algorithms unambiguously. Moreover, with the public release of the fingerprint databases, FVC inspired a lot of young researchers to explore and make advancements in the field.

The main objective of this project is to analyze the Fingerprint Recognition problem as an Image Processing and Pattern Recognition problem and to consider the various techniques and practices proposed, in order to develop a AFRS.

The reminder of this report is organized as follows. Section 2, discusses the possible ways of representing the fingerprint image. Section 3, explains the fingerprint segmentation process. Section 4, elaborates the implementation of different image enhancement techniques. Section 5, discusses the process of feature extraction from the fingerprint images. Section 6 and 7, explain the process of fingerprint matching and evaluate the performance of the system developed. Finally, section 8 indicates some deficiencies of the system and proposes techniques which can be used for its improvement. Although, the process of Fingerprint Acquisition is an important stage of a real-world AFRS, due to its hardware nature it was not studied during the course of this project.

2. Fingerprint Representation

The main characteristic of every AFRS is the selection of the fingerprint pattern representation. This characteristic of the system strongly influences the selection of image enhancement and feature extraction techniques. For this reason, it is very important in the very beginning to distinct between the different representations of fingerprint images and to define the one used in the AFRS developed.

In (Maltoni et al., 2009), depending on the different scales of analysis and types of features, the fingerprint pattern representations are structured in the following 3 levels:

- Level 1, or *global level*, examines the line flow of the ridges. Singular points, called *loops* and *deltas* (fig. 1a) are identified and act as control points around which all other ridges are wrapped. These features may be used for fingerprint indexing or classification, but are not sufficient for fingerprint matching.
- Level 2, or *local level*, identifies the local ridge characteristics. Most common ridge characteristics called *minutiae* are: *ridge endings*, and *ridge bifurcations* (fig. 1b). Fingerprint minutiae are persistent and remain unchanged over an individual's lifetime. For this reason, the minutiae are the most commonly used fingerprint representation.
- Level 3, or *very fine level*, detects *intra-ridge details*. These include width, shape, curvature, and edge contours of the ridges. One of the most important intra-ridge details are the fingerprint *swear pores* (fig. 1c), whose positions and shapes are considered to be highly distinctive. However, the techniques for extracting level 3 features are only applicable to images with very high quality and are less usable in practice.

Figure 1, visually illustrates each of the fingerprint representations discussed. During the development of the AFRS for this project the level 2, minutiae based representation was considered as most suitable and was adopted.

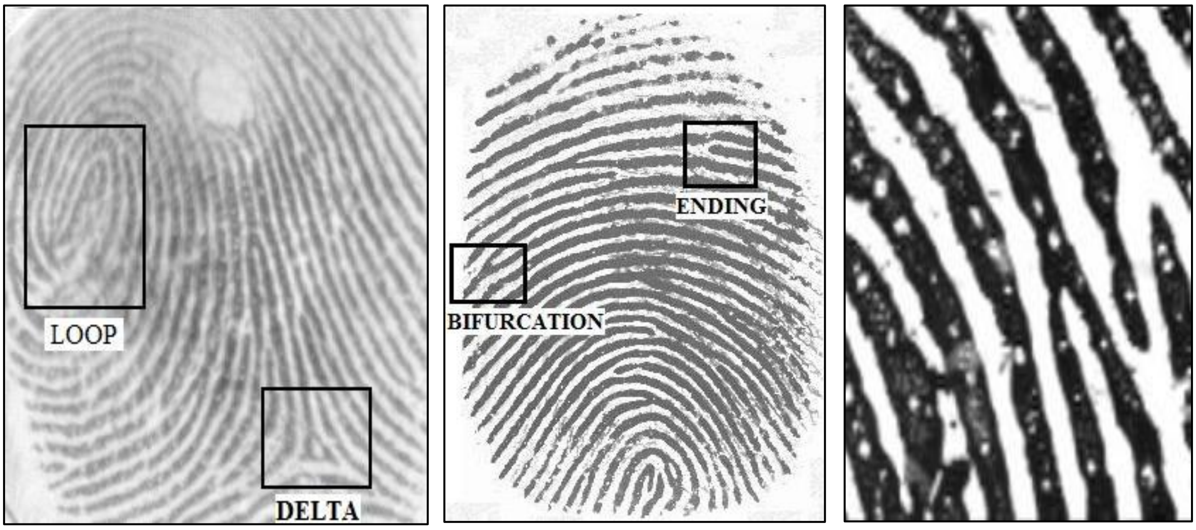

Figure 1: a) Level 1 features, loop and delta; b) Level 2, minutiae, ridge endings and bifurcation; c) Level 3, swear pores (a – FVC2000, DB2, FP Impressions No. 103-2; b, c - (Maltoni et al., 2009))

3. Image Segmentation

The first stage after the fingerprint acquisition is the process of fingerprint segmentation. *Segmentation* refers to the process of separation of the fingerprint area (foreground) from the image background. Since the background region usually includes noise patterns, segmentation is useful to avoid the extraction of spurious features.

Due to the nature of the fingerprint images a simple global or local intensity thresholding techniques does not provide efficient results. However, many other fingerprint segmentation techniques have been proposed in the literature. (Mehtre et al., 1987) proposed isolating the foreground region based on local histograms of fingerprint orientation, where the orientation of each pixel and the histogram of every 16x16 window are computed. The local histograms with significant peaks are considered as a foreground, whereas the flat or near-flat histograms are considered as background. (Bazen and Gerez, 2001) used a learning based segmentation where the gradient coherence, intensity mean, and intensity variance are used as input features in a liner classifier to distinct between the background and foreground pixels.

For the purpose of this project a simpler and more intuitive technique proposed by (Mehtre, 1993) was adopted. This technique is based on the assumption that due to the ridge-valley nature of the fingerprint image, the foreground regions have higher grayscale variance than the background regions. First the global grayscale variance of the image is computed, then the image is divided in blocks of size $B \ x \ B$ and the grayscale variance of each block is computed. If the local variance is less than the global variance, than the block region is assigned to the background and to the foreground otherwise. The following formula was used to calculate the local grayscale variance:

$$V(k) = \frac{1}{B^2} \sum_{x=0}^{B-1} \sum_{y=0}^{B-1} (I(x,y) - M(k))^2$$

where *V(k)* is the grayscale variance of the block *k*, *I(x, y)* is the grayscale intensity of the pixel at *(x, y)*, and *M(k)* is the mean intensity of the block *k*.

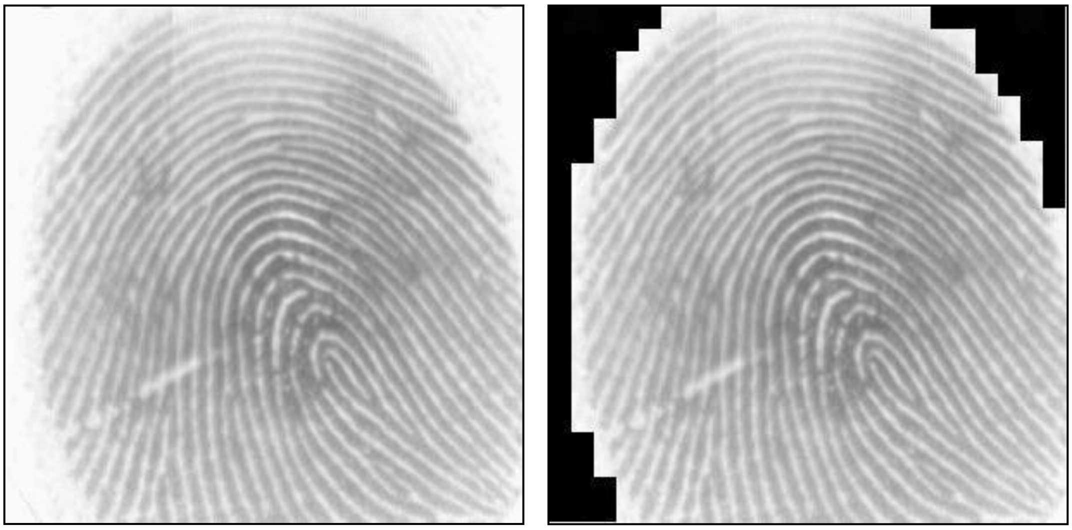

Figure 2: Image Segmentation based on local variance thresholding, B = 10
(Original image taken from: FVC2000, DB2, FP Impression No. 105-8)

Figure 2, illustrates the operation of the segmentation algorithm. The image on the left is the original fingerprint image and the image on the right shows the result of the segmentation algorithm with block size *B=10*, where the background region is colored black for more effective illustration.

4. Image Enhancement

The quality of the ridge structure of the fingerprint image is essential for successful feature extraction. Ideally, the ridges and valleys in the image should alternate and flow in constant direction. However, in practice, due to the presence of noise the ridge structure is not always well defined. Thus, various image enhancement techniques are used in order to reduce the noise and stress the definition of ridges against valleys. The selection of image enhancement techniques is strongly influenced by the fingerprint representation adopted by the system. For this reason, only techniques which facilitated the process of minutiae extraction were considered. Various fingerprint enhancement techniques, both in the special and frequency domains has been proposed in the literature. Some of them include: pixel-wise enhancement, contextual filtering, multi-resolution enhancement, crease removal, and other. For the purpose of this project, the widely adopted technique proposed in (Hong, Wan and Jain, 1998), was considered as most suitable. This technique includes convoluting the image with a *Gabor filter* adjusted to the local frequency and orientation of the fingerprint image. The main stages include: normalization, orientation estimation, frequency estimation, and filtering. Figure 3 depicts the operation of the Gabor Filter in terms of input and output.

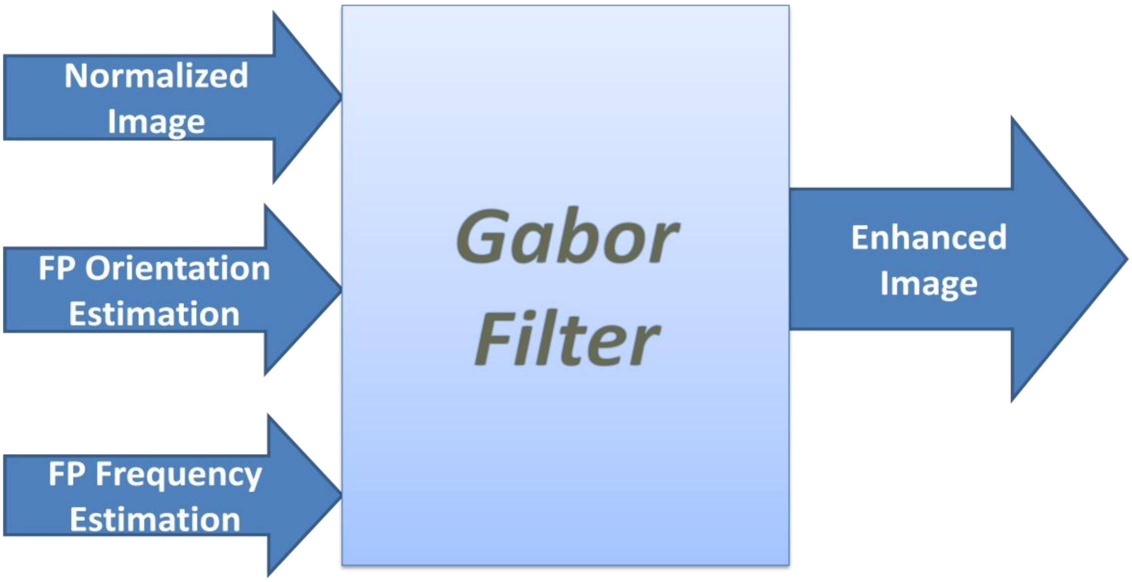

Figure 3: The operation of the Gabor Filer

4.1 Normalization

The first stage of this enhancement technique involves normalization of the fingerprint image. This ensures that the image has a specified mean and variance. Due to the non uniform contact with the sensing device the fingerprint image may have distorted level of variation along the ridges and valleys. Normalization reduces the distortion effects and facilitates the enhancement process in the subsequent stages. As suggested in (Hong, Wan and Jain, 1998) the intensity value in the normalized image is computed as follows:

$$I'(x,y) = \begin{cases} M_0 + \sqrt{\dfrac{V_0 * (I(x,y) - M)^2}{V}} & if\ I(x,y) > M, \\ M_0 - \sqrt{\dfrac{V_0 * (I(x,y) - M)^2}{V}} & otherwise, \end{cases}$$

where, *I'(x, y)* denotes the intensity in the normalized image at *(x, y)*, *M* and *V* denote the mean and variance of the image, M_0 and V_0 denote the desired mean and variance respectively. Figure 4, illustrates the operation of the normalization algorithm.

Figure 4: Fingerprint image before and after normalization, V_0 = 0, M_0 = 1
(Original image taken from: FVC2000, DB2, FP Impression No. 107-6)

4.2 Orientation Estimation

The ridge orientation at a pixel *(x, y)* is the angle θ_{xy} that the fingerprint ridges, crossing through an arbitrary small neighborhood centered at *(x, y)*, form with the horizontal axis (Maltoni et al., 2009). Effective estimation of the ridge orientation is crucial step, since the Gabor filter in the subsequent steps relays on the orientation to successfully enhance the image. The *least mean square estimation algorithm* proposed in (Hong, Wan and Jain, 1998) was employed for computing the orientation image. The algorithm includes the following steps:

1) Dividing the normalized image in blocks of size *B x B*,
2) Computing the gradients Δx and Δy using the *Sobel masks* (Gonzalez and Woods, 2008),
3) Estimating the local orientation, $\theta(i,j)$ at each block centered at *(x, y)* calculated using the following equations:

$$V_x(i,j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\Delta x(u,v) * \Delta y(u,v),$$

$$V_y(i,j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (\Delta x^2(u,v) - \Delta y^2(u,v)),$$

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

4) Due to the presence of noise in the input image this results may not always be the best estimates. For this reason, local low-pass filter is used to modify the incorrect orientation estimates. However, prior to performing the filtering the orientation image is converted into a *continuous vector field* as follows:

$$\Phi_x(i,j) = \cos\big(2\theta(i,j)\big),$$
$$\Phi_y(i,j) = \sin\big(2\theta(i,j)\big),$$

where Φ_x is the x component and Φ_y is the y component.

5) The *Gaussian low-pass filter* of size *5x5* is applied to the continuous vector and the filtered Φ'_x and Φ'_y components are computed,

6) Finally, the smoothed orientation estimation is defined as follows:

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

Figure 5, depicts the operation of the orientation estimation algorithm.

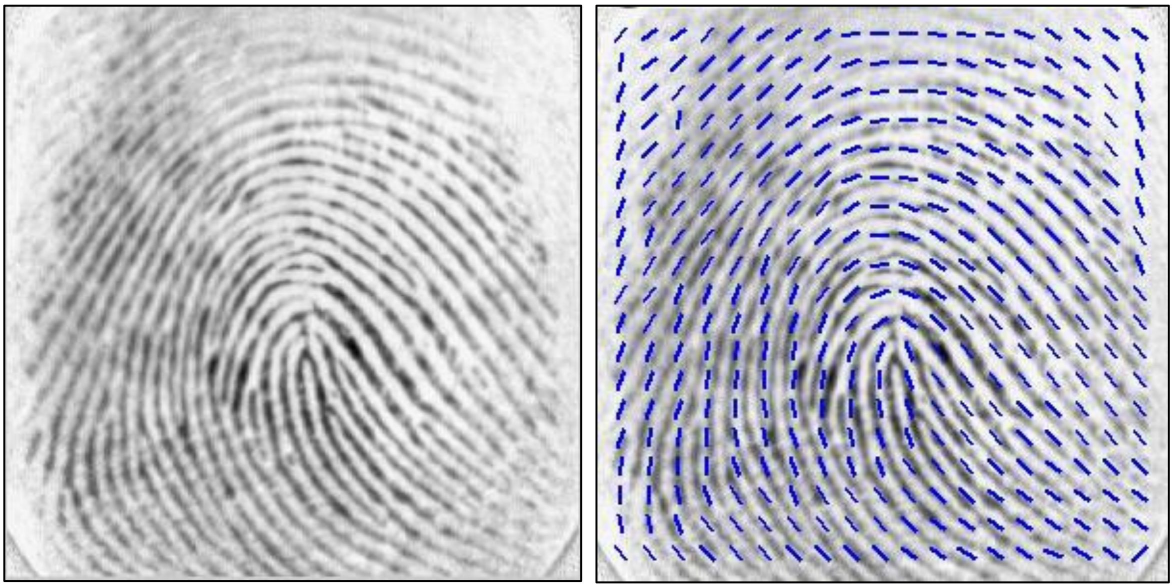

Figure 5: Fingerprint orientation estimation, a) Normalized image; b) Orientation image (Original image taken from: FVC2000, DB2, FP Impression No. 107-6)

4.3 Ridge Frequency Estimation

Another important parameter used in the construction of the Gabor filter is the local ridge frequency estimation. The initial step in the computation of the ridge frequency is dividing the image in blocks of size *B x B*. Next, the grayscale values of all pixels located inside the block along the orientation orthogonal are projected. This projection has a sinusoidal form where the local minima correspond to the ridges in the fingerprint. Next, the ridge spacing *S(x, y)* is computed by counting the median number of pixels between consecutive minima points. Finally, the ridge frequency for a block centered at *(x, y)* can be computed as:

$$F(x, y) = \frac{1}{S(x, y)}$$

However, this technique fails to estimate the ridge frequency when there are minutiae in the block or there are no consecutive peaks in the projection. For this reason, the out of range frequency values are interpolated from the frequency of the neighboring blocks which have well-defined frequency.

4.4 Filtering

After the ridge orientation and ridge frequency estimations have been computed, these parameters are used to construct the *Gabor filter*. As explained in (Daugman, 1985), Gabor filters have both *frequency-selective and orientation-selective properties* and have optimal joint resolution in both spatial and frequency domains. For this reason, the Gabor filters are suitable for removing the noise in the fingerprint images while preserving its structure. As defined in (Daugman, 1985), the even-symmetric Gabor filter has the following general form:

$$G(x, y : \Phi, f) = \ exp\left\{-\frac{1}{2}\left[\frac{(x\cos\Phi)^2}{\delta_x^2}+\frac{(y\sin\Phi)^2}{\delta_y^2}\right]\right\}\cos(2\pi\ fx\ cos\Phi),$$

where, Φ is the orientation of the Gabor filter, f is the frequency of a sinusoidal plan wave, and δ_x and δ_y are the space constants of the *Gaussian envelope* along x and y, respectively.

As mentioned previously, in the case of fingerprint enhancement, Φ corresponds to the local ridge orientation estimation, and f corresponds to the local ridge frequency estimation. The selection of the δ_x and δ_y parameters involves a trade-off. Larger values results in filters which are more robust to noise, but may create spurious ridges. On the other hand, smaller values may be less effective in reducing the noise, but are less likely to create spurious ridges. As suggested in (Hong, Wan and Jain, 1998) the values of δ_x and δ_y were set to 4. Finally, to obtain the enhanced image the normalized image is spatially convoluted with the above defined filter. As shown in figure 6, the results of the enhancement are impressive.

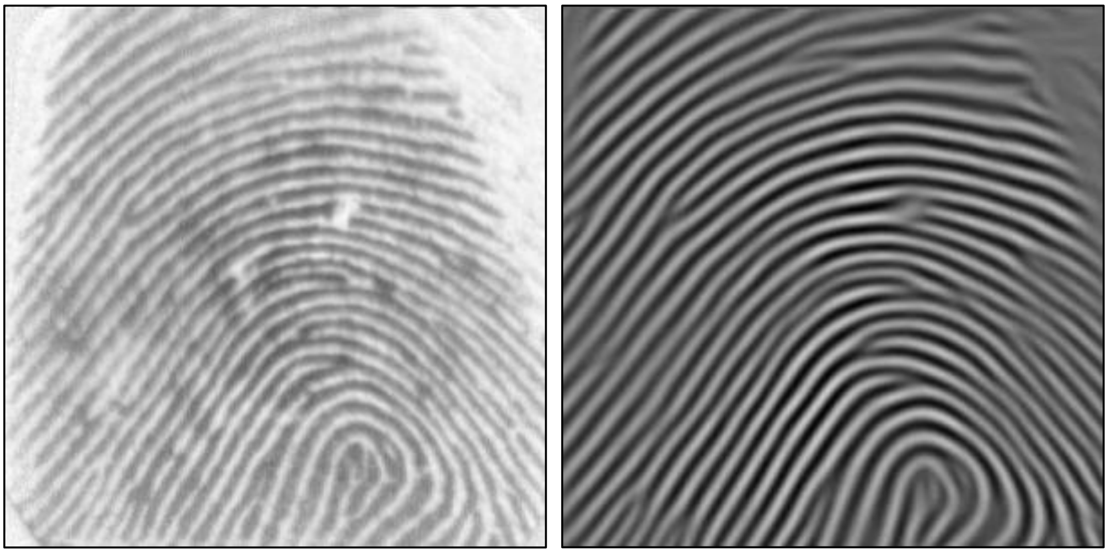

Figure 6: Fingerprint Image Enhancement with Gabor Filter, a) Original image; b) Filtered image (Original image taken from: FVC2000, DB2, FP Impression No. 101-5)

4.5 Binarization

Binarization refers to the process of conversion from grayscale to binary images. The main reason for binarization of the fingerprint images is that most minutiae extraction algorithms operate on binary images. Furthermore, binarization improves the contrast between the ridges and valleys in the fingerprint image and further facilitates the process of feature extraction.

Two methods for binarization have been considered: *binarization using global and local threshold*. The binarization with global threshold includes calculating the mean grayscale intensity of the image and setting this value as a global threshold *t*. Thus, every pixel in the image with intensity level lower than *t* is assigned a value of 0, and all other pixels a value of 1. On the other hand, binarization with local threshold includes dividing the image in blocks of size *B x B*, calculating the mean grayscale intensity of each block, and using this value as a local threshold *t*. Next, each pixel in the block is assigned a value 1, if its intensity exceeds the local threshold *t*, and 0 otherwise. In general, different portions of the image may characterize with different contrast and intensity, thus in theory the binarization with local threshold is suspected to produce more satisfying results. However, when applied to the fingerprint images from database used for this project, the global binarization method showed significantly better results. As shown in the figure, the binarization using local threshold introduces spurious connections between the fingerprint ridges, which create difficulties in the process of minutiae extraction. On the other hand, the binarization with global threshold clearly separates the ridges and stresses the valleys between them.

Figure 7: Binarization, a) Enhanced Fingerprint Image; b) Binarization using local threshold, with block size B=20; c) Binarization using global threshold (Original image taken from: FVC2000, DB2, FP Impression No. 105-2)

4.6 Skeletonization

The final stage of preprocessing the fingerprint images prior to minutiae extraction is skeletonization. *Skeletonization* refers to the process of thinning the foreground regions (fingerprint ridges) in the binary fingerprint image until they are one pixel wide. In order to obtain suitable results, an algorithm which guarantees maximally thin, connected, and minimally eroded skeleton is needed. One elegant technique for obtaining the skeletonized image is *morphological skeletonization.* This technique defines the image skeleton in terms of successive morphological erosions and openings (Gonzalez and Woods, 2008). However, morphological skeletonization makes no provisions for keeping the skeleton connected and thus is not suitable for skeletonization of fingerprint images. Therefore, more sophisticated skeletonization algorithm explained in (Gonzalez and Woods, 2008) was adopted. The method consists of successive passes of two basic steps applied to the boarder points of the foreground regions. Each of these steps analyzes the eight-neighborhood of the border points and according to predefined set of rules flags border points for deletion. Thus, one iteration of the skeletonizing algorithm consists of applying the first step to flag border point for deletion, deleting the flagged points, applying the second step to flag the remaining points for deletion, and finally deleting the flagged points. This procedure is applied iteratively until no further points are flagged for deletion, when the algorithm terminates, yielding the skeletonized image. The effects of the algorithm are illustrated in the figure below.

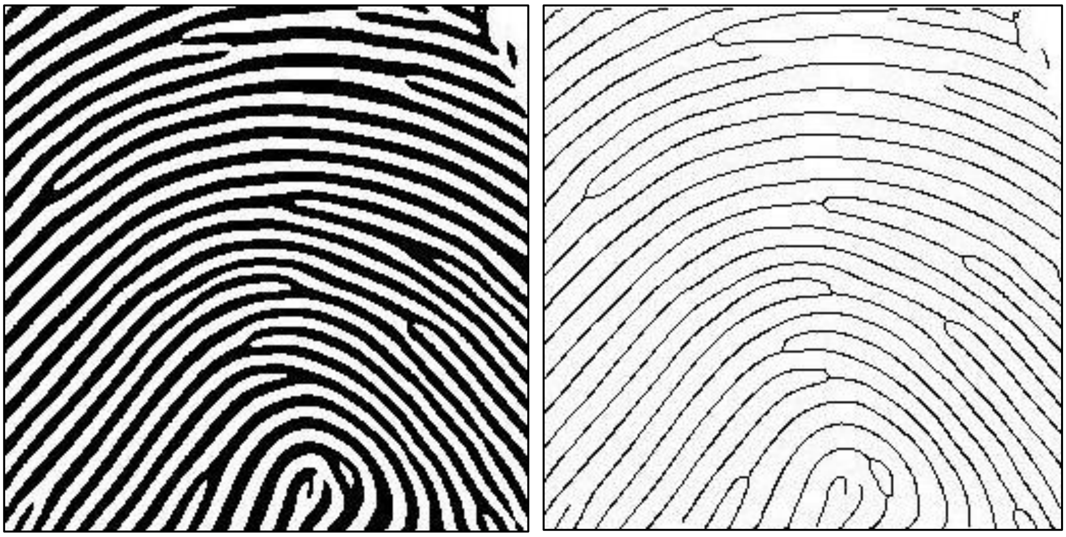

Figure 8: Skeletonization, a) Binarized fingerprint image; b) Skeletonized image (Original image taken from: FVC2000, DB2, FP Impression No. 101-2)

5. Minutiae Extraction

Once all the preprocessing steps are applied and the binary skeleton of the fingerprint image is obtained the next step is to extract the minutia. The minutiae extraction was performed by using the widely adopted concept of *Crossing Number* (CN). This technique allows all minutiae in the fingerprint image to be extracted by only a simple image scan of the binary skeleton image. As defined in (Maltoni et al., 2009), the Crossing Number *CN(p)* of a foreground pixel p is defined as the half-sum of the differences between pairs of adjacent pixels in the eight-neighborhood of p:

$$CN(p) = \frac{1}{2} \sum_{i=1 \ldots 8} |val(p_{i \bmod 8}) - val(p_{i-1})|$$

where p_0, p_1, ... p_7 are the pixels belonging to an ordered sequence of pixels defining the eight-neighborhood of the pixel p, and *val(p)* is the pixel value which has to belong to the set {0, 1}. By using the properties of the crossing number each ridge pixel can be classified as an intermediate point (non-minutiae), ridge-ending, or bifurcation. For example, a ridge pixel with CN number of two corresponds to intermediate point, CN number of one corresponds to ridge-ending, and CN number of three corresponds to bifurcation. Figure 9, depicts these properties of the CN number.

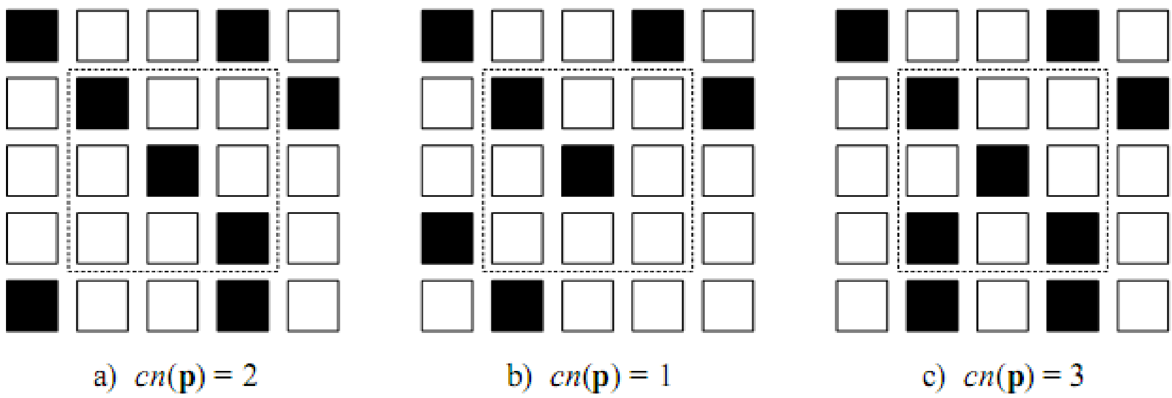

Figure 9: Crossing Number; a) Intermediate point; b) Ridge ending; c) Bifurcation (Maltoni et al., 2009)

Once, all minutiae points are detected a set of *(x, y, θ, CN)* quadruples are stored for each fingerprint image. Where *(x, y)* are the spatial coordinates of each minutiae point, *θ* is the orientation angle in *(x, y)* as defined in section 2.2, and *CN* is the crossing number of the minutiae. As it will be explained in the next section, these sets are further used in the process of matching the fingerprint images. Figure 10, illustrates the use of the *CN* for minutiae extraction. All ridge pixels marked with red rectangles have *CN* value of one or three and correspond to minutiae points.

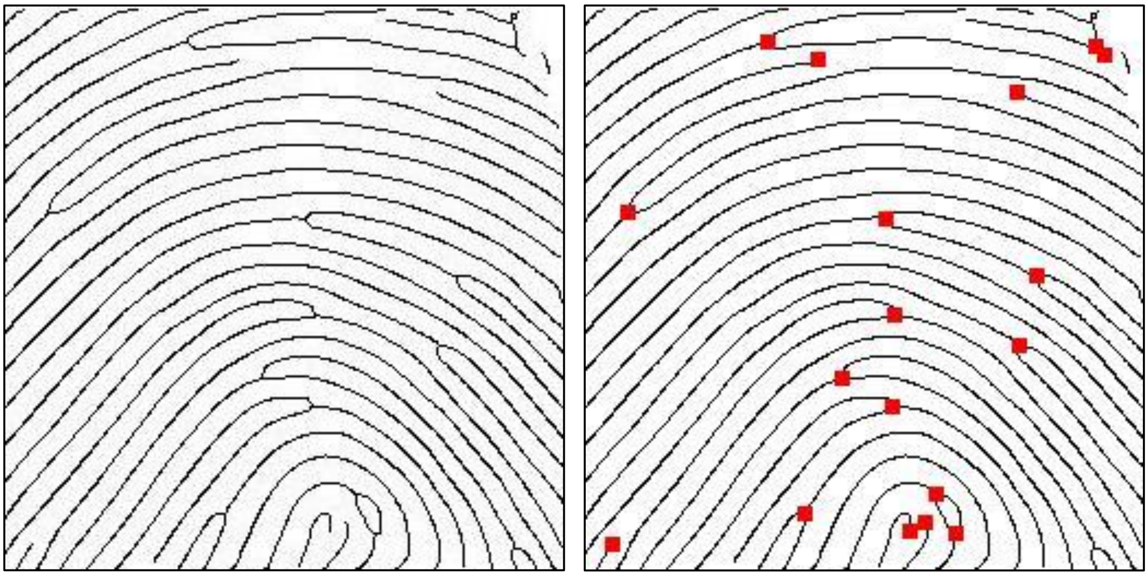

Figure 10: Minutiae Extraction using Crossing Number
(Original image taken from: FVC2000, DB2, FP Impression No. 101-2)

6. Fingerprint Matching

The purpose of fingerprint matching algorithms is to compare two fingerprints and to return their degree of similarity. Due to the large variability in different impressions of the same finger, the problem of fingerprint matching is very challenging. An effective matching algorithm has to cope with the displacement, rotation, non-linear distortion, and noise in the fingerprint images. In (Maltoni et al., 2009), the matching algorithms are classified in three categories: Correlation-based, Minutiae-based, Non-Minutiae feature-based. *Correlation-based* methods operate by superimposing the images and computing the correlations between the images for different alignments. *Minutiae-based* methods essentially consist of finding the alignments of the fingerprint images which results in maximum number of minutiae pairings. *Non-Minutiae feature-based* methods operate by extracting other features in the fingerprint patters and using these features compute the correlation between different fingerprints.

For the purpose of the AFVS developed for this project the minutiae-based fingerprint matching algorithm proposed in (Ratha et al., 1996) was adopted. Namely, the algorithm consists of three steps: registration, minutiae-pairing, and matching score computation. *Registration* refers to the process of finding the 'best' transformation which when applied to one of the minutiae sets results in maximum number of overlapping minutiae. Finding this transformation requires estimation of the translation, rotation, and scaling parameters. Thus, for each of the parameters a discretized set of allowed transformations is defined and the matching score is calculated. The matching scores are collected in accumulation array *A*, where each entry counts the evidence of a given translation, rotation, and scale parameters. For each minutiae pair (p, q), where p is point in feature set *P*, and q is point in the set *Q*, all possible transformations that map p to q are found, and the evidence for these transformations are incremented in the array A. For each pair of scaling and rotation values (s, θ) there is only one shift vector $(\Delta x, \Delta y)$ such that $T_{(s,\, \theta,\, \Delta x,\, \Delta y)}(p) = q$, and can be found as:

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = q - s \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} p$$

The values $\Delta x, \Delta y$ are quantized and the evidence is added to the array *A*. This procedure is repeated for all scale and rotation values for all minutiae pairs in the sets *P* and *Q*.

Finally, the transformation with maximum score in *A* is believed to be the correct one. The algorithm is summarized in the figure below.

<u>Registration</u>:
A(k, l, m, n) = 0, k=1, …, K; l = 1, …, L; m = 1, …, M; n = 1, … , N
FOR $(p_x, p_y, \alpha) \in P$:
  FOR $(q_x, q_y, \beta) \in Q$:
    FOR $\theta \in \{\theta_1, \ldots, \theta_M\}$:
      IF $\alpha + \theta = \beta$:
        FOR $s \in \{s_1, \ldots, s_N\}$:
          $\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = q - s \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} p$
          Add evidence in A for $T_{\theta, s, \Delta x, \Delta y}$
RESULT = $\mathrm{argmax}_{k,l,m,n}$ A(k,l,m,n)

Figure 11: Fingerprint Matching: Registration Algorithm

The next step is *minutiae pairing*. Two minutiae are considered as paired or matched, if their components *(x, y, θ)* are equal after registration. To add flexibility to the minutiae pairing procedure a *tolerance box* (Δx, Δy, $\Delta\theta$) is defined. Thus, if the difference between the minutiae components is within the range defined by the tolerance box, they are still considered as matched. In order to reduce computations, for each minutia set a *bounding box* is defined. The bounding box is smallest rectangular region which encloses all minutiae points. Thus, the minutiae points which do not belong to the intersection of the bounding boxes (referred as common bounding box) of the two minutes sets are not considered for pairing.

Once, the minutiae points are registered and paired, the last step is to calculate the *matching score*. The matching score *MS* of two minutiae sets (fingerprints) *p* and *q*, is defined as follows:

$$MS(p,q) = \frac{m^2}{(n_p * n_q)}$$

where, m is the number of minutiae paired, n_p is the number of minutiae in p belonging to the common bounding box, and n_q is the number of minutiae in q belonging to the common bounding box. The matching score is the output of the matching algorithm and represents the degree of similarity between the two fingerprints.

7. Evaluation of the AFVS Developed

Once, all the components of the AFVS were implemented the system was evaluated. For the purpose of the evaluation the DB2 Set B from the Fingerprint Verification Competition 2000 was used. The database subset is publicly available through the official website of the competition. It is 10 fingers wide and 8 impressions per finger deep, i.e. 80 fingerprints in total. The fingerprint images have been acquired by using a low-cost capacitive sensor and each image is of size 256 x 364. Each of the fingerprint images was preprocessed as explained in the previous sections, and the minutiae set was extracted and stored for each fingerprint. Further, these minutiae sets were used as an input to the matching algorithm to evaluate the system as defined below.

In (Maio et al., 2000), there are strictly defined measures for evaluating the AFVSs. These measures are widely adopted in research community and used for evaluating the systems during the Fingerprint Verification Competitions. The performance measures are defined as follows:

- Each fingerprint sample is matched against the remaining samples of the same finger and these scores are stored as ***gms*** - *Genuine Matching Scores*. If the matching *p* against *q* is performed, the symmetric one (i.e., *q* against *p*) is not executed to avoid correlation. The number of matches performed, ***NGRA*** – *Number of Genuine Matching Attempts* is ((8*7) / 2) * 100 = 280.
- Each fingerprint sample is matched against the first sample of the remaining fingers in database and the scores are stored as ***ims*** – *Impostor Matching Scores*. If the matching *p* against *q* is performed, the symmetric one (i.e., *q* against *p*) is not executed to avoid correlation. The number of matches performed, ***NIRA*** – *Number of Impostor Matching Attempts* is ((10*9) / 2) = 45.
- The ***FMR(t)*** - *False Match Rate*, and ***FNMR(t)*** **-** *False Non-Match Rate* curves are computed from the above distributions, where t is between 0 and 1. Given a threshold t, *FMR(t)* denotes the percentage of $ims \geq t$, whereas *FNMR(t)* denotes the percentage of $gms < t$, i.e:

$$FMR(t) = \frac{\{ims|ims \geq t\}}{NIRA}, \quad FNMR(t) = \frac{\{gms|gms < t\}}{NGRA}$$

- The ***EER*** - *Equal Error Rate* is computed as the point where FMR(t) =FNMR(t). This score gives a single value which shows the overall performance of the AFVS.

The chart below shows the *FMR* and *FNMR* curves computed during the evaluation of the system, and defines the *EER* at **35%**. Since the system was evaluated only at the subset of the fingerprint database, it is ungrateful to compare its performance to the winning system in the competition which had much lower *EER* of 0.61%.

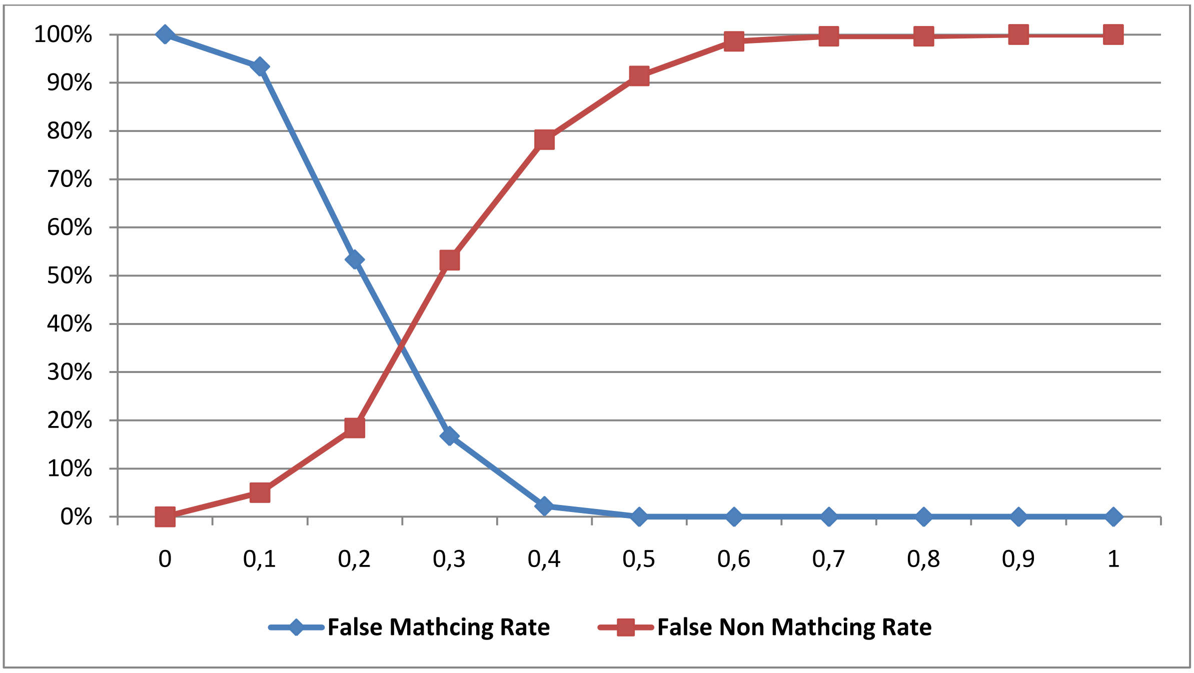

Figure 12: The performance of the AFVS developed

8. Conclusions and Future Work

The primary focus of this project was to experiment with the various Image Processing and Pattern Recognition techniques proposed for building a AFVS. Throughout the project a series of Image Processing techniques for fingerprint image enhancement were discussed and implemented. Moreover, the various methods for representing the fingerprint images were considered. Primary attention was paid to the minutiae-based technique for feature extraction. Furthermore, a fingerprint matching algorithm based on minutiae matching was discussed and implemented. Finally, the different measures for evaluating the AFVSs were studied and used to evaluate the system implemented.

However, due to the short time frame in which the project had to be pursued some steps which may significantly improve the performance of the system were skipped. Some of them are:

- *Filtering Extracted Minutiae* – Due to the low quality of some segments of the fingerprint image they cannot be recovered even after the applying the various enhancement techniques. Consequently, these segments result in extracting a large number of spurious minutiae points. For this reason, a technique for filtering the minutiae extraction is needed.
- *Fingerprint Classification* – In the current state, in order the system to answer whether a given fingerprint is already present in the database, the query fingerprint image has to be matched against all fingerprint images in the database. This requires a lot of time and processing. Fingerprint Classification refers to the process of classifying the fingerprint images into a number of classes which are further used for indexing. Thus, when given a query fingerprint it is matched only with fingerprint images belonging to the same class. This will significantly reduce the number of matchings and the response time of the system.

9. References

(Bazen and Gerez, 2001) Bazen, M. and Gerez, H. (2001) *Segmentation of Fingerprint Images*, Workshop on Circuits Systems and Signal Processing.

(Daugman, 1985) Daugman, G. (1985); *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters*, J Opt Soc Am A.

(Gonzalez and Woods, 2008) Gonzalez, R. and Woods, R. (2008) *Digital Image Processing*, 3rd edition, New Jersy: Pearson Prentice Hall.

(Hong, Wan and Jain, 1998) Hong, L., Wan, Y. and Jain, A. (1998), *Fingerprint Image Enhancement: Algorithm and Performance Evaluation*, IEEE Transactionson Pattern Analysis and Machine Intelligence, 777–789.

(Leung et al., 1991) Leung, W.F., Leung, S.H., Luk, A. and Lau, W.H. (1991), *Fingerprint Recognition Using Neural Network*, *Neural Networks For Signal Processing*, IEEE Workshop.

(Maio et al., 2000) Maio, D., Maltoni, D., Cappelli, R., Wayman, J. and Jain, A. (2000) *FVC2000: Fingerprint Verification Competition*.

(Maltoni et al., 2009) Maltoni, D., Maio, D., Jain, A. and Prabhakar, S. (2009) *Handbook of Fingerprint Recognition*, 2nd edition, London: Springer.

(Mehtre, 1993) Mehtre, M. (1993), *Fingerprint image analysis for automatic identification*, Machine Vision and Applications, 124-139.

(Mehtre et al., 1987) Mehtre, M., Murthy, N., Kapoor, S. and Chatterjee, B. (1987) *Segmentation of fingerprint images using the directional image*, Pattern Recognition, 429–435.

(Ratha et al., 1996) Ratha, N., Karu, K., Shaoyun, C. and Jain, A. (1996), *A Real-time Matching System for Large Fingerprint Databases*, Pattern Analysis and Machine Intelligence, 799 - 813.