Sabanci University

TE 407

Digital Image Processing

Final Project Report:

“Filterbank-Based Fingerprint Matching”

June 28, 2004

Didem Gözüpek & Onur Sarkan

5265 5241

1. Introduction

The need for security in personal identification increases because of the new means of communication like e-commerce, electronic banking systems, automatic teller machines, smart cards, and cellular telephones, automatic personal identification applications are becoming an indispensable necessity. However, traditional personal identification methods like passwords and personal identification numbers (PINs) are prone to fraud.
Therefore, more reliable automatic personal identification methods are becoming vital.

Biometrics refers to identifying an individual based on his or her physiological or behavioral characteristics. A biometrics based system can be operated in two modes: verification mode and identification mode. The former either accepts or rejects a user's claimed identity, whereas the latter establishes the identity of the user without a claimed identity information. Among many biometrics based methods such as face recognition, iris scan, speech recognition, hand geometry; fingerprint recognition is one of the most mature and proven techniques.

A fingerprint is a pattern of ridges and valleys. The uniqueness of a fingerprint can be detected by the overall pattern of ridges and valleys, which constitutes the global information, or the local ridge anomalies. The popular fingerprint representation schemes are either based on predominantly local landmarks or on exclusively global information. One example for the purely local information based techniques is the *minutiae* (ridge branches and ridge ends) based matching systems. However, different fingerprints have different number of minutiae; which makes these systems extremely difficult in implementation since it is not suitable for indexing mechanisms. On the other hand, systems based on exclusively global information such as correlation based systems which match the global pattern of ridges and valleys to determine if the ridges align, are not very reliable because it is very probable

that different fingerprints have the same global configuration. Therefore, a method which utilizes both global and local information in the identification of a fingerprint can produce very good results.[1]

2. Method

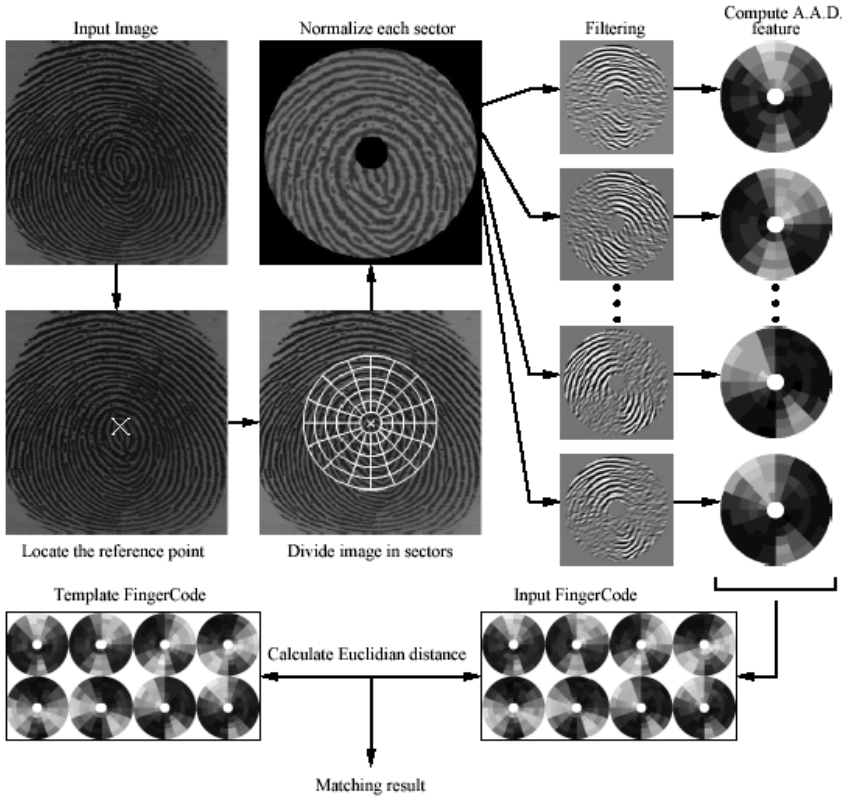
The method that has been proposed combines the local and global information in a fingerprint via a feature extraction scheme and represents this in a short, fixed-length code, called *FingerCode*, and implements the matching phase by finding the Euclidean distance between these fingercodes.

The main parts of the algorithm are as follows:

1. Determine a reference point and region of interest for the fingerprint image
2. Locate the region of interest (ROI) around the reference point.
(sectorization & normalization)
3. Filter the ROI using a bank of Gabor filters that are tuned in 8 directions.
4. Compute the Average Absolute Deviation (AAD) of gray values in individual sectors in filtered images to define the feature vector, the *FingerCode.*
5. Implement the matching by finding the Euclidean distance.

This algorithm can be illustrated as follows:

[1] Anil K. J., Anil, P. Salih, P. Sharath, H. Lin, "Filterbank-based Fingerprint Maching" ,IEEE Transactions on Image Processing, Vol. No.5, May 2000.



2.1. Reference Point Location

The reference point is the point of maximum curvature of the concave ridges in a fingerprint. The algorithm is based on the analysis of the orientation fields. Orientation field, *O(i,j)* for a fingerprint image, is a PxQ image representing the local ridge orientation at pixel (i,j). Local ridge orientation is usually specified for a block

rather than at every pixel. Hence, the image is divided into blocks of size w x w and a single orientation is defined for each block:

1. Divide I , the input image, into non-overlapping blocks of size w x w.
2. Compute the gradients ∂_x (i,j)and ∂_y (i,j) at each pixel (i,j).
3. Estimate the local orientation of each block centered at pixel (i,j) using:

$$\mathcal{V}_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v)$$

$$\mathcal{V}_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v))$$

$$\mathcal{O}(i,j) = \frac{1}{2} tan^{-1}\left(\frac{\mathcal{V}_y(i,j)}{\mathcal{V}_x(i,j)}\right)$$

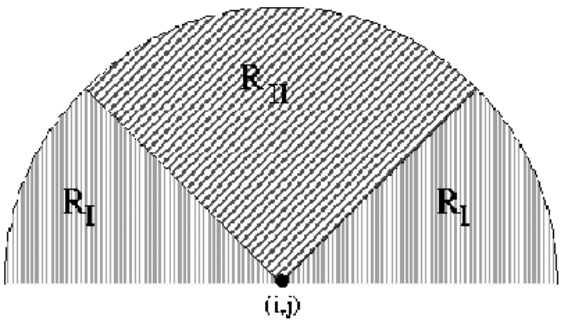
where O(i,j) is the least square estimate of the local ridge orientation at the block.

4. Smooth the orientation field in a local neighborhood (low pass filtering).

5. Compute the image, containing only the sine component of the smoothed orientation field.

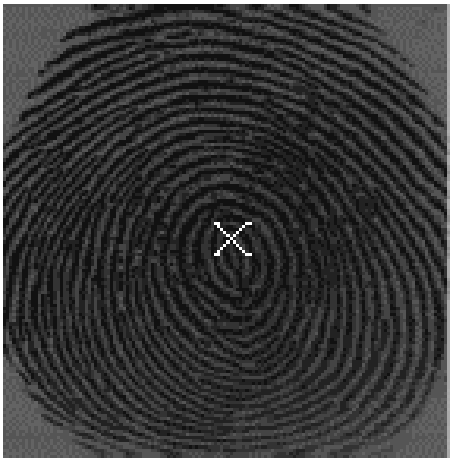
$$\mathcal{E}(i,j) = sin\left(\mathcal{O}'(i,j)\right)$$

6. For each pixel in $\mathcal{E}$; i.e., sine component of the smoothed orientation field, integrate pixel intensities in regions RI and RII, and assign the correponding pixels in A, the value of their difference:



7. Find the maximum value in A and assign its coordinate to the reference point.

The reference point for an image should be as follows:



2.2. Sectorization

The region of interest is the collection of all the sectors Si such that:

$$\begin{aligned} S_i &= \{(x,y)\,|b(T_i+1) \le r < b(T_i+2), \\ &\quad \theta_i \le \theta < \theta_{i+1}\ , 1 \le x \le N, 1 \le y \le M\} \end{aligned}$$

where

$$\begin{aligned} T_i &= i\ div\ k, \\ \theta_i &= (i\ mod\ k) \times (2\pi/k), \\ r &= \sqrt{(x-x_c)^2+(y-y_c)^2}, \\ \theta &= tan^{-1}((y-y_c)/(x-x_c)) \end{aligned}$$

The sectorized image should be as follows:

2.3. Normalization

The aim of normalization is to remove the effects of sensor noise and gray level deformation due to fingerprint pressure differences. The mathematical model for this procedure is as follows:

$$N_i(x,y) = \begin{cases} M_0 + \sqrt{\frac{V_0 \times (I(x,y)-M_i)^2)}{V_i}}, & if \quad I(x,y) > M_i \\ \\ M_0 - \sqrt{\frac{V_0 \times (I(x,y)-M_i)^2)}{V_i}}, & otherwise, \end{cases}$$

where

I (x,y) = gray value at pixel (x,y)

M_i = estimated mean of sector Si

V_i = estimated variance of sector Si

M_0= desired mean value (we chose it to be 100)

V_0= desired variance value (we chose it to be 100)

2.4. Gabor Filterbanks

Gabor filters are a well-known technique to capture useful information in specific bandpass channels. For example, a fingerprint convolved with a 0^0 oriented Gabor filter accentuates the ridges that are parallel to the x-axis, and smoothes the ridges in other directions.

An even symmetric Gabor filter has the following form in the spatial domain:

$$G(x, y; f, \theta) = exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\} cos(2\pi f x')$$
$$x' = xsin\theta + ycos\theta,$$
$$y' = xcos\theta - ysin\theta,$$

where

f = frequency of the sinusoidal plane wave along the direction from the x-axis.

∂_x and ∂_y = space constants of the Gaussian envelope.

While Gabor filters tuned to 4 directions can capture the global information in a fingerprint, 8 directions can capture the local information as well. Minutiae points are the

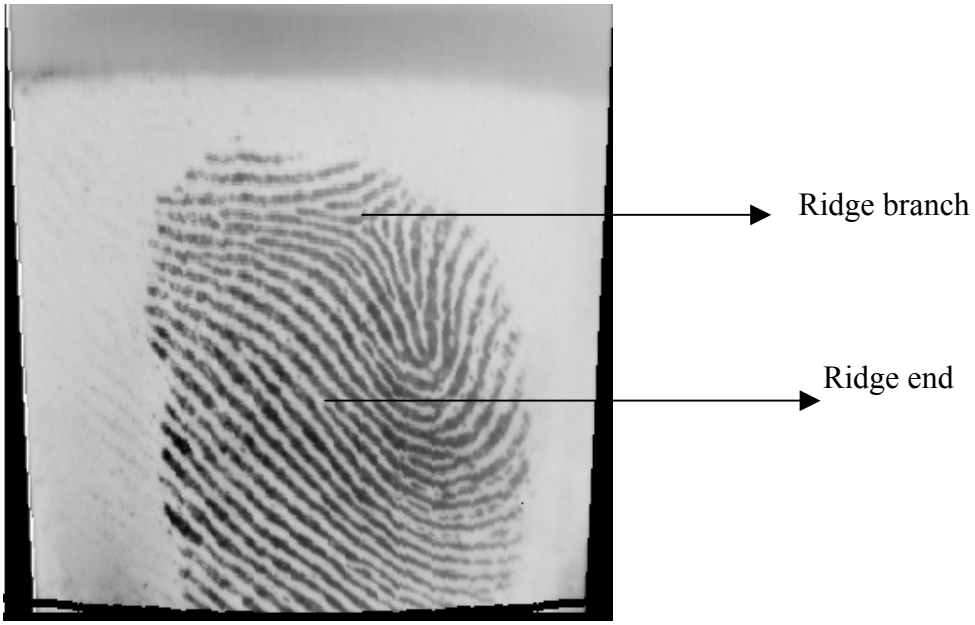
Local ridge anomalies and it is this information that we want to capture using Gabor filters.

Hence, filters tuned to

0^0, 22.5^0, 45^0, 67.5^0, 90^0, 112.5^0, 135^0, and 157.5^0 are implemented.

To recall, minutiae points are the ridge branches and ridge ends, which may be

illustrated as follows:



2.5. Feature Vector

Let $F_{i\theta}(x,y)$ be the θ-direction filtered image for sector S_i. For all i ->{1,2,…,80},
and θ->{0^0, 22.5^0, 45^0, 67.5^0, 90^0, 112.5^0, 135^0, 157.5^0 },the feature value $V_{i\theta}$; which is the average absolute deviation from the mean (AAD) is:

$$V_{i\theta} = \frac{1}{n_i}\left(\sum_{n_i} |F_{i\theta}(x,y) - P_{i\theta}|\right)$$

where

n_i = number of pixels in sector S_i

$F_{i\theta}$ = mean of pixel values of $F_{i\theta}(x,y)$ in sector S_i .

The $V_{i\theta}$ for each sector in the Gabor filtered images constitute the feature vector, FingerCode.

2.6. Matching

Fingerprint matching is based on finding the Euclidean distance between the corresponding FingerCodes. Translation invariance is achieved by the reference point, whereas the rotation invariance is achieved by cyclically rotating the features in the FingerCode itself.

3. Implementation

To implement the above mentioned algorithms, we wrote following seven functions based on filter-bank based fingerprint matching method: findCenter, sec_norm_image, gaborfilters, feature_vector, matching, who_is_this, and training functions.

findCenter:

This function find the point of maximum curvature of the concave ridges in the fingerprint image. This point is the reference point to compare different fingerprint image.

sec_norm_image:

This function produce suitable pre-processed fingerprint image for Gabor Filters. In the filter-bank based fingerprint matching method, according to reference point, region of interest (region of characteristic ridges and valleys) are divided several sectors according to pixel location (distance and angle with refence point). This mean that we need a lot of calculations. Instead of these calculations, we produced sectorizing map. This is 121x121 matrix, which have 80 sectors around the center. This sectorization map is saved in the matlab directory as a mat file.

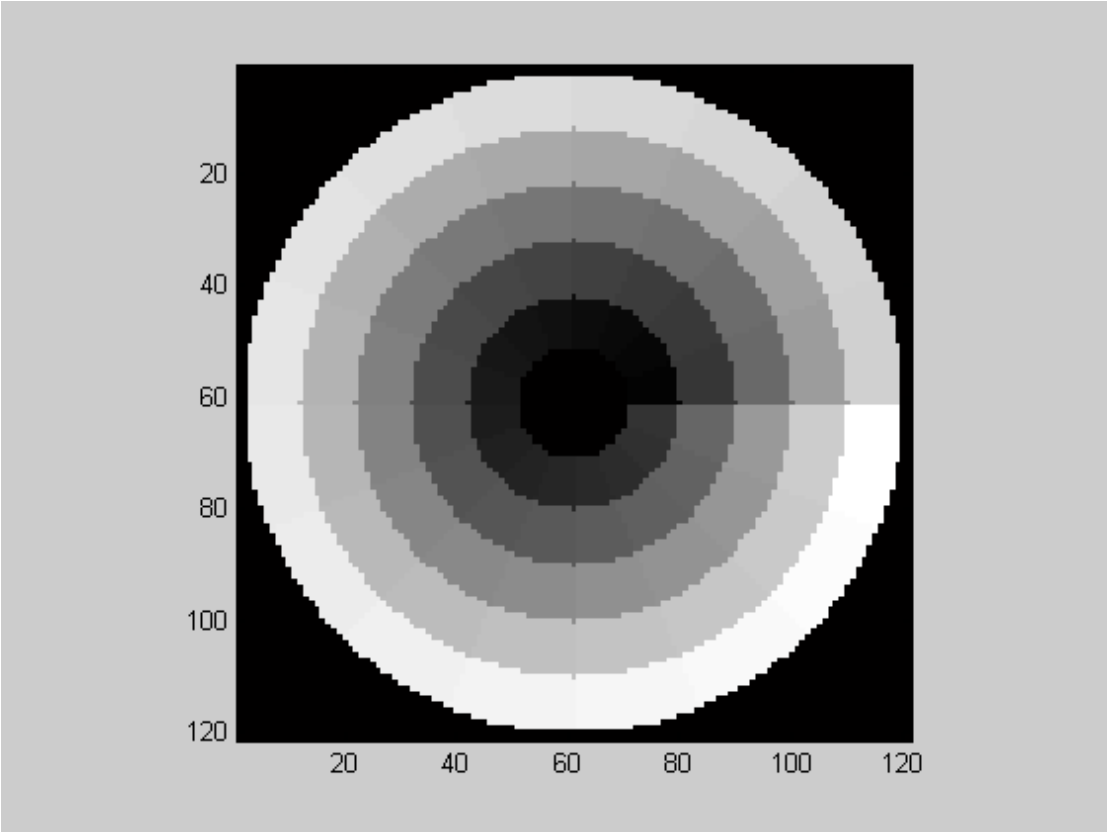
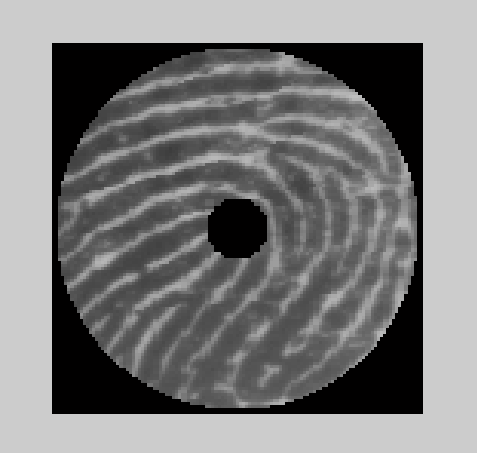


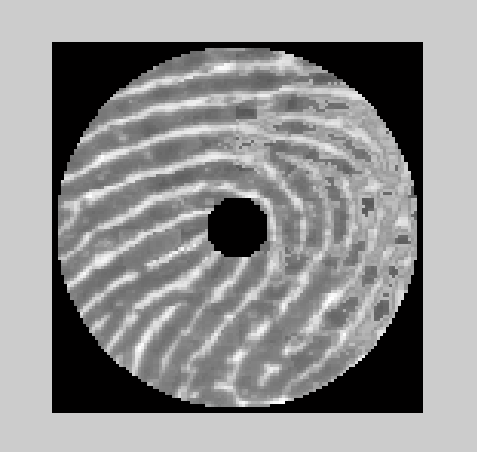
Figure 1 (Sectorization Map)

During the sectorization,we load this matrix. We know coordinates of reference point. We paste sectorized map on reference point, and crop region

of interest from the fingerprint image. We produced 121x121x2 matrix(region of interest + sectors). This is the example for a sectorized image:



Because of different ligthing and pressure, brightness intensity can be different for fingerprint images of same person. Gabor filter results are easily effected by this difference. To protect results from this difference, we normalized sectorized fingerprint images in this function. Normalization is separately performed for each sector of image. The desired mean and variance values were chosen to be both 100. Darker images become lighter, lighter images become darker. Following image is the normalized form of previous image:



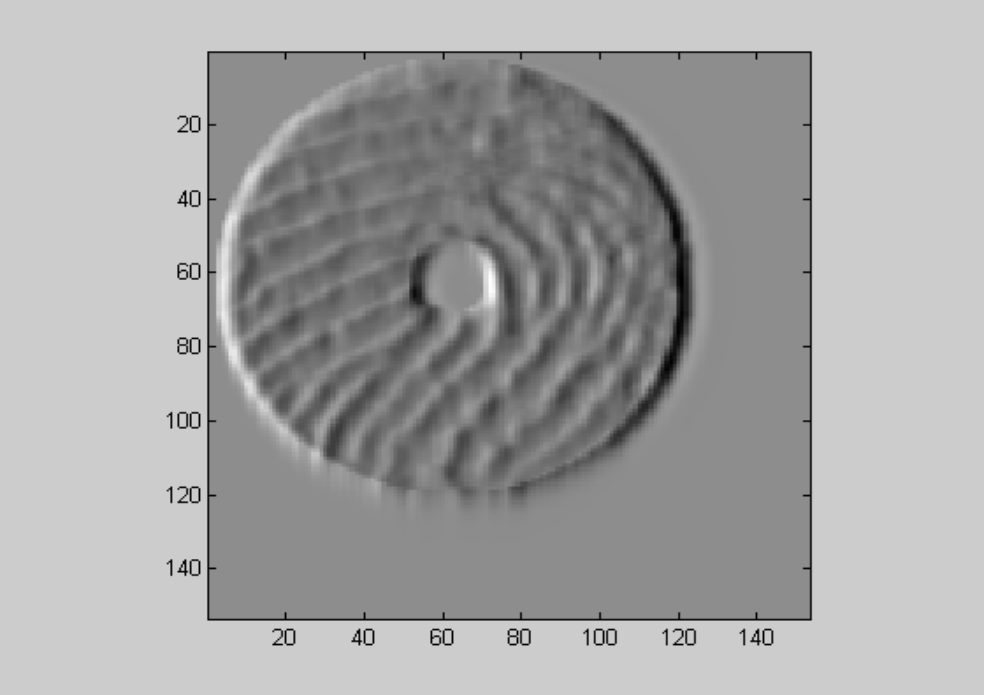
Output of function is sectorized and normalized 121x121 image.

gaborfilters:

To capture spesific information in spesific direction, we apply 8 different gabor filter on the sectorized and normalized image. We just implement following formula of even symetric gabor filter.

$$
\begin{aligned}
G(x, y; f, \theta) &= exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\} cos(2\pi f x'), \\
x' &= xsin\theta + ycos\theta, \\
y' &= xcos\theta - ysin\theta,
\end{aligned}
$$

We chose f =1/10 because average distance between parallel ridges and valleys is about 10 pixels. This is the average period of the ridges and valleys. By this function we produce 8 different 33x33 gabor filter kernels. And we convolve sectorized and normalized images with these kernels. Output of each convolution is 153x153 image. We combine eigth 153x153 matrix in the 153x153x8 matrix. This matrix is the output of the function. Following images is one example of the results of gabor filtering.



feature_vector:

This function takes result of gabor filters(153x153x8). It find mean of each of 80 sectors in each gabor filtered image. This process is applied for each of 8 segments(results of 8 different gabor filters.). Results are saved into 80x8 matrix. This is the feature matrix of fingerprint image.

matching:

This function takes 80x8 feature matrix of image, and calculate distance between base fingerprints' feature vectors, and fingerprint is matched according to this distance.

who_is_this:

This is the main function of this project. This function takes fingerprint image name. It reads this image from directory. It find its reference point by findCenter function. By sec_norm_image function, it sectorized and normalized input image. By gaborfilters function, it applies 8 different gabor

filters on sectorized and normalized image. By feature_vector function, it find feature matrix of input image. Finally, by matching function, it finds the nearest base sample in database, and according the distance decides whether it matches or not.

Results & Conclusion

This filterbank-based fingerprint matching technique utilizes both the local and global information in a fingerprint image, hence it consists of the advantages of both methods.

Moreover, the FingerCode requires only about 640 bytes of storage depending on the size of the fingerprint image. Hence, it is also attractive in terms of its memory allocation.

However, one bottleneck of the algorithm is that the reference location algorithm fails on poor quality images, which is in fact a very difficult task to achieve.

Actually, we have tried to implement this system in OpenCV. However, there was a bug which we could not fix and we had little time; therefore, we switched to implementing it in Matlab.

As a result, 7 out of 8 people were identified. Although this might seem a small percentage at first glance, since our database was small and the unidentified image was a bad image, this percentage does not completely reflect the performance.

In terms of execution speed, the algorithm works well since the average time for identifying an image is about 4 seconds. However, if a larger database were used, it would take more time (hence slower) to search the database.

As a further improvement, Gabor filters in 16 directions might be implemented since it would capture more local informaiton. Hence, a performance comparison in terms of accuracy, execution speed etc., can be made, which would provide us more insight in the implementation and improvements of the algorithm in real-life implementation.