Fingerprint Recognition System for Affine Transformed Fingerprints

Sundar Ram
Department of Electronics and Computer Engineering
Drexel University
Philadelphia, PA
ss3782@drexel.edu

Fernand Cohen
Department of Electronics and Computer Engineering
Drexel University
Philadelphia, PA
fscohen@coe.drexel.edu

***Abstract*— Fingerprint recognition systems have been very important in all formats of life, school, college, offices, banks, etc. There has been a lot of different types of fingerprint recognition algorithms, but not many affine invariant algorithms are very powerful. In this paper, I have developed a method which to recognize fingerprints that have also undergone affine transformations.**

Keywords-affine invariant; fingerprint recognition; computer vision; minutia points

I. Introduction

Fingerprints are the oldest and most widely used form of biometric identification. The two fundamental premises on which fingerprint identification is based are: 1) Fingerprint details are permanent and 2) fingerprints of an individual are unique [1]. As most Automatic Fingerprint Recognition Systems are based on local ridge features known as minutiae, marking minutiae accurately and rejecting false ones is very important. However, fingerprint images get degraded and corrupted due to variations in skin and impression conditions. Thus, image enhancement techniques are employed prior to minutiae extraction. A critical step in automatic fingerprint matching is to reliably extract minutiae from the input fingerprint images. A fingerprint is a unique pattern of ridges and valleys on the surface of a finger of an individual. A ridge is defined as a single curved segment, and a valley is the region between two adjacent ridges. Minutiae points, as shown in Fig.1, are the local ridge discontinuities, which are of two types: ridge endings and bifurcations [2]. These minutiae points are used for determining uniqueness of a fingerprint.

This is a project in which there are 10 fingerprints in a database and 5 test fingerprints are given. We are required to match the test fingerprints with the exact ones in the database. We are also required to find the error of the test fingerprints with respect to all the fingerprints in the data base.

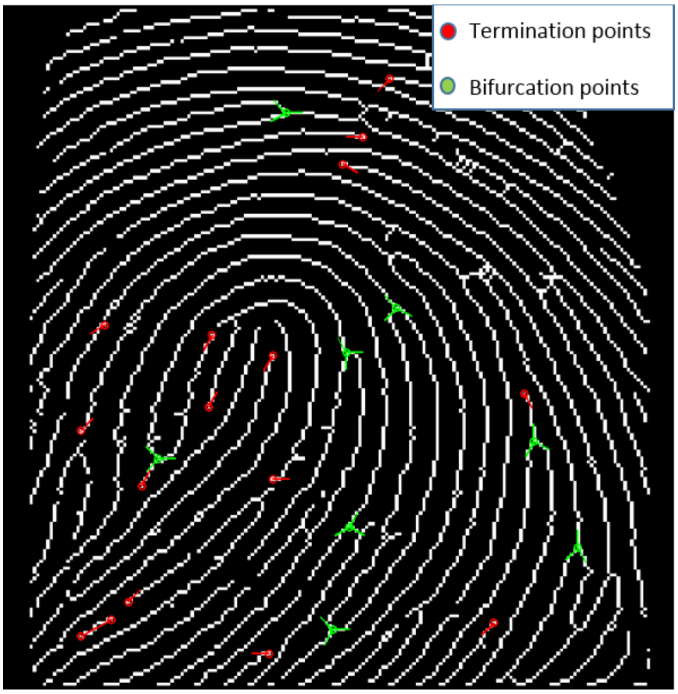

Fig.1 Representation of Minutiae points. Green circles represent bifurcation points and red circles represent termination points. The lines from the circles represent their orientations.

II. Methodology

A. Preprocessing

A critical step in automatic fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint images. However, the performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. In order to ensure that the performance of an automatic fingerprint identification/verification system would be robust with respect to the quality of the fingerprint images, it would be essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module.

The first step is to binarize the image, meaning convert the given grayscale, color or any image to logical arrays by using Otsu's thresholding algorithm. Vary the threshold to correctly math your requirements. The threshold I chose in my project was 1.1 times the output of the Otsu thresholding algorithm. After the operation, ridges in the fingerprint are highlighted with black color while furrow are white.

Since the lines on the fingerprints are really wide, it is very difficult to find the termination and bifurcation points. So we have to thin the ridges. Ridge thining is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide.

B. Minutiae Extraction

Now, we are ready to extract the minutia points. We filter the thinned ridge map by the filter "minutie". "minutie" computes the number of one-valued elements in each 3x3 window:

if the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a termination.

if the central pixel is 1 and has 3 one-value neighbors, then the central pixel is a bifurcation.

if the central pixel is 1 and has 2 one-value neighbor, then the central pixel is a normal pixel.

Now, we are having a lot of spurious or unwanted minutiae. We must remove all the spurious minutiae. To remove the spurious minutiae, what we decided to do was first to find the distance between the minutiae points already found. If the distance is very close to a threshold value, then, remove that minutia. So, the permutations here to be considered are the distance between two bifurcation points, distance between two termination points and the distance between the bifurcation and the termination points. For this project, my threshold turned out to be 6 pixels.   I have attached the database and test set of fingerprints in Fig. 2 and Fig. 3. They would give a clear picture of what is the motivation and the variables involved in calculation of the invariance.

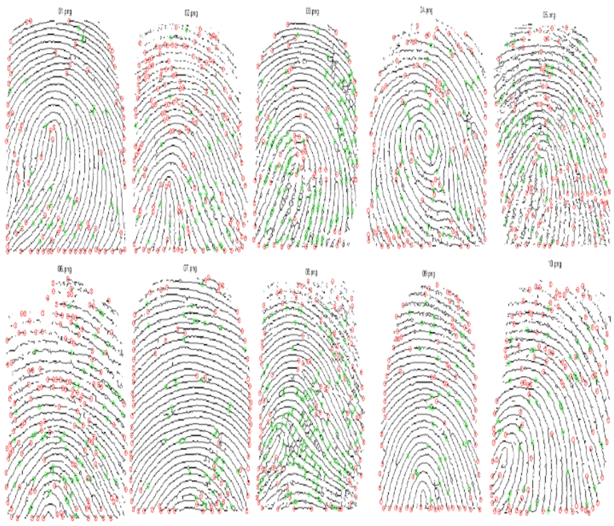

Fig. 2. Database of the finger prints. All the fingerprints have the minutia marked in them. Red points represent termination points and green represents bifurcation points.

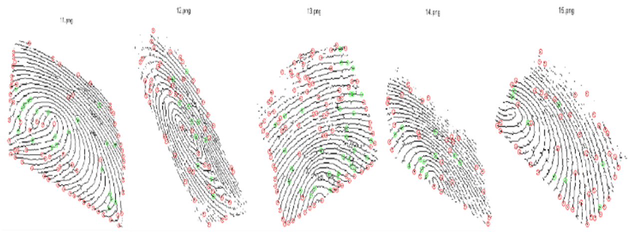

Fig. 3. Test fingerprints. All the fingerprints have the minutia marked in them.

III. MATCHING THE FINGERPRINTS

Now comes the most difficult task. To match the correct fingerprint. Let me explain the steps in detail. For affine transformation, we know that the ratio of the areas of the triangles are same before and after transformation. So, our aim is to find the ratio of the triangles. But, for that, how do we get the triangles? We are first plotting the convex hulls of the extreme minutia points. Then finding the ratio of the areas of the consecutive minutia points. For a curve with n minutia points, let the sequence of areas be denoted by (A_1,A_2,A_3,. …,A_{n-2}). Let the area ratios for the original one be given by the equation:

$$I = \{I(k) = A_k/A_{k+1}\,, k = 1,2,3, \ldots, n-3\}. \tag{1}$$

Let the ratio of the transformed image be given by the equation:

$$I' = \{I'(k) = A_k/A_{k+1}\,, k = 1,2,3, \ldots, n-3\}. \tag{2}$$

And we have,

$$I' = I. \tag{3}$$

We allow for a small error (2-5%) in the values of the invariants, to declare matching points. The points are declared matching, if

$$|I'(j) - I(i)| < 0.05 \cdot I(i) \tag{4}$$

for every pair in the two sets.

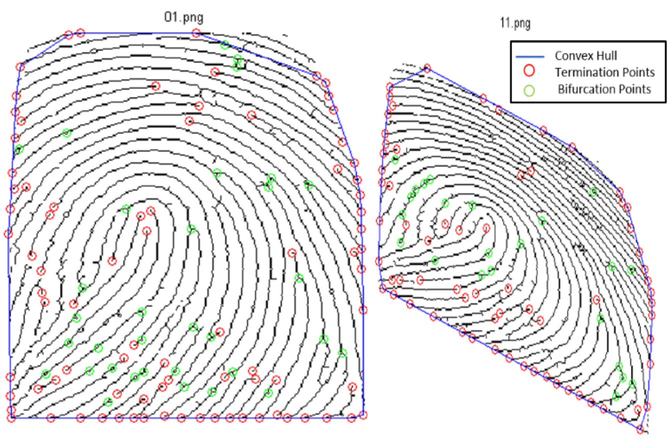

Fig. 4. Example of Convex Hull plotting with extreme minutiae points.

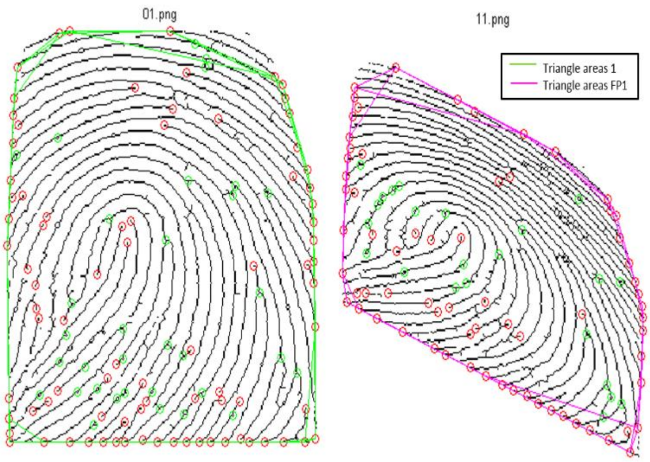

Fig. 5. Example of finding the areas of consecutive triangles in database image and test image.

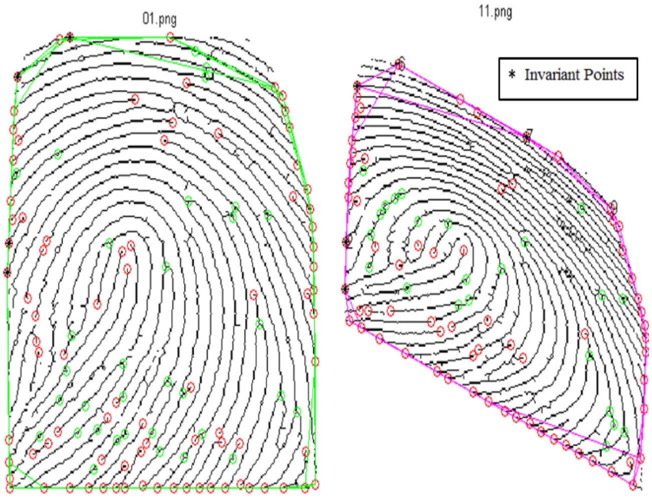

Fig. 6 The invariant points are marked with black asterisk and the points are also numbered on the transformed image.

Once we have the matching points, then we have to find the [L] matrix or the affine matrix. We already know that,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (5)$$

So, we have to place all the transformed points on the left hand side of the equation and all the original coordinates on the right side of the equation. Implement a pseudo-inverse and solve for the [L] matrix. In this project, b matrix is zero. There is no translation. So b is taken care of already.

There are 5 test fingerprints. Find the matching invariance points with all the 10 database points. Once you find [L] matrix for the test fingerprint 1, multiply this [L] matrix with the minutia points in all the database points. Calculate the mean square error between the test fingerprint and the transformed versions of the database images. But how do we know which are the corresponding points? For this, we use the k-nearest neighbours approach. We calculate the error between the minutia points in the transformed database images and the test image. The least mean square error will give the correct matching fingerprint.

ACKNOWLEDGMENT

I thank Dr. Cohen for giving us a project which is very well used in real life scenario and making us believe that the systems out there are not really difficult to make. The project was really interesting. I also thank Zexi for constantly guiding us through the project without any delay and being so lenient in correcting our assignments and quizzes.

REFERENCES

[1] R. M. Bolle, A. W. Senior, N. K. Ratha and S. Pankanti "Fingerprint minutiae: A constructive definition." Biometric Autentication (2002) 58-62.

[2] Roli Bansal, Priti Sehgal and Punam Bedi, "Minutiae Extraction from Fingerprint Images - a Review", International Journal of Computer Science Issues, Vol. 8, No. 5, 2012, pp. 74-85.

RESULTS

From the table shown below, we see that FP1 = 1, FP2 = 4, FP3=6, FP4 = 9, FP5=10.

ERROR	1	2	3	4	5	6	7	8	9	10
FP1	6.761198	12.23292	18.73946	10.16635	11.2592	10.44138	14.05587	11.85247	18.38135	15.14628
FP2	31.2654	25.06301	38.41907	21.69333	26.90604	28.49242	33.44927	27.15521	33.31911	29.35415
FP3	19.64832	11.69547	15.7226	14.05739	14.69979	11.30969	20.80179	13.76682	16.83516	12.79274
FP4	23.37351	12.98871	19.18045	16.02535	13.16844	10.82802	26.09399	20.74906	8.141972	17.12329
FP5	38.48155	35.24376	45.03941	32.88175	34.88581	32.6475	36.58182	30.65072	37.34364	26.67719

Table 1. Mean square error between test and database fingerprints. Highlighted pixels indicate least error.

BLOCK DIAGRAMS