ISSN No. 0976-5697

Volume 4, No. 10, September-October 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Hough Transform Based Fingerprint Matching Using Minutiae Extraction

Nitika Saroha*
M.Tech Student of Department of computer Science and Application,
Maharshi Dayanand University,
Rohtak, Haryana-124001, India
nitisuhag@gmail.com

Nasib Singh Gill
Department of computer Science and Application,
Maharshi Dayanand University,
Rohtak, Haryana-124001,
India
nasibgill@gmail.com

Abstract: Fingerprints are the most widely used biometric feature for identification and verification in the field of biometrics. This paper is focused on developing a system for recognizing two fingerprints using minutiae matching and Hough transform. Fingerprint matching is the process of comparing a fingerprint against another fingerprint to determine if the impressions are from the same finger. The paper proposes a new scheme to improve the accuracy of fingerprint recognition. Fingerprint Matching using minutiae extraction includes The matching using displacement and the rotation. The Hough transform is introduced to include the scaling. This proposed scheme matches the fingerprint by using Hough transform and the minutiae matching.

Keywords: Hough Transform, Minutiae Extraction, Fingerprint Matching, FMR

I. INTRODUCTION

The fingerprint of an individual is unique and is formed from an impression of the pattern of ridges and valley on a finger. A ridge can be defined as a single curved segment, and a valley can be defined as the region between two adjacent ridges (Figures 1-3).A minutiae refers to the local discontinuities in the ridge flow pattern and provides the features that can be used for biometric identification. The characteristics such as orientation and location of minutiae are usually taken into account when performing fingerprint matching. A typical example of fingerprint that represents ridges, valleys and minutiae is shown in Figures 1-3 below.

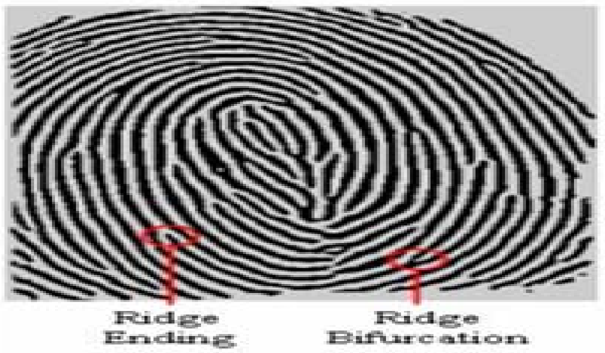


Figure 1: An example of fingerprint image showing ridges ending and bifurcation.

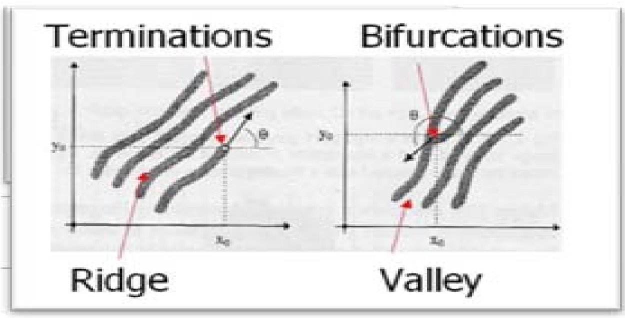


Figure: 2 An example of ridges and valley in fingerprint image

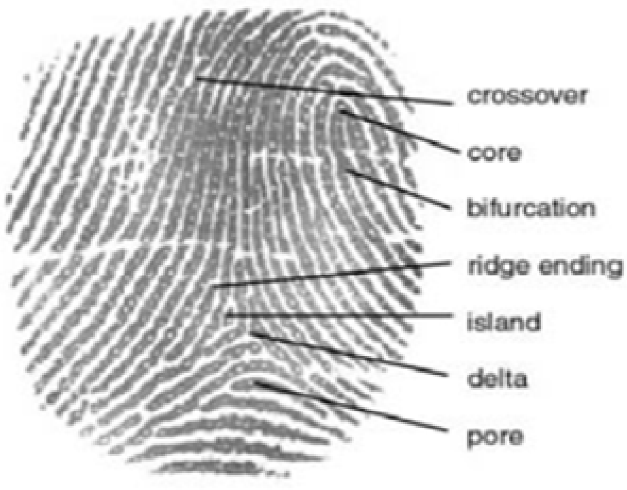


Figure: 3 Key minutiae feature

A. *Fingerprint Recognition:*

Fingerprint recognition is one of the reliable, important and very useful biometric techniques used for person verification and identification [1]. The term recognition is used in a general sense and comprise of all three kinds of tasks i.e. Verification (or authentication), Identification and Classification.

a. A verification system authenticates a person's identity by comparing the captured fingerprints with her own biometric template(s) pre-stored or already in the system. It follows one-to-one comparison to determine whether the identity claimed by the individual is true [10];
b. An identification system recognizes an individual by searching the entire template database for a fingerprint matching. It follows one-to-many comparisons to establish the identity of the individual [2].

The verification is much easier and faster because here we have the two fingerprints i.e. input fingerprint with its own fingerprint pre-stored in the system and we just need to compare them. On the other hand, the identification implies more time for extracting the fingerprint because there are needed much more details [4]. Minutiae based fingerprint recognition consists of binarization; Thinning, Minutiae extraction; Minutiae matching and lastly computing matching score [8].

Motivation: The motivation behind the work is growing need to identify a person for security. A fingerprint matching is one of the popular biometric methods used to authenticate human being. The proposed fingerprint matching provides reliable and better performance than the existing technique.

Contribution: In this paper we used Fingerprint Recognition using Hough transform and minutiae extraction method with the help of MATLAB codes. The minutiae are extracted from the thinned image for both the input and the template image. Finally both the images are subjected to matching process and matching score is computed.

II. HOUGH TRANSFORM

The simplest case of Hough transform is the linear transform for detecting straight lines. In image space, the straight line can be described as $y = mx + b$ where the parameter m is the slope of the line, and b is the intercept i.e. (y-intercept). This can be defined as the slope-intercept model of a straight line. In case of Hough transform, a main idea is to consider the characteristics of the straight line not as discrete image points (x1, y1), (x2, y2), but in a terms of its parameters according to the slope-intercept model. In general, the straight line $y = mx + b$ can be represented as a point (b, m) in the parameter space. But, vertical lines pose a problem. They are more naturally described as x = a and would give rise to unbounded values of the slope parameter m. Hence, for computing, Duda and Hart proposed the use of a different pair of parameters, denoted as a and θ, for the lines in the Hough transform. Both of These two values define a polar coordinate.

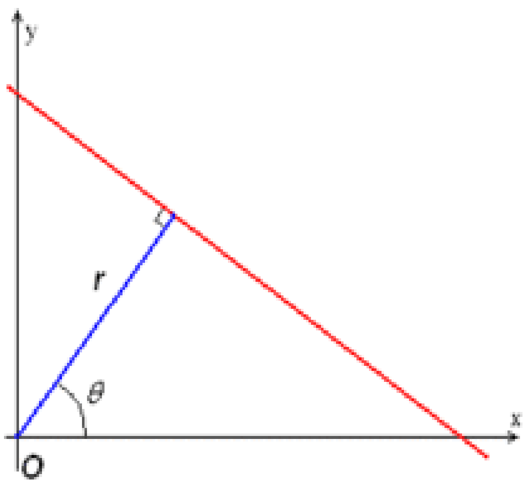


Figure 4 Slope intercept of straight line

The parameter represents the distance between the line and the origin, and theta is the angle of the vector from the origin to this closest point (see Coordinates). By using this parameterization, equation of the line can be written as [7].

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

Which can be rearranged to $r = x\cos\theta + y\sin\theta$.

A. *Hough Transform Implementation:*

The linear Hough transform algorithm uses a two-dimensional array, called an accumulator, to detect the existence of a line described by $r = x\cos\theta + y\sin\theta$. The dimension of the accumulator equals the number of unknown parameters, i.e., two, considering quantized values of r and θ in the pair (r, θ). For each pixel at (x, y) and its neighborhood, the algorithm of Hough transform determines if there is enough evidence of a straight line at that particular pixel. If so, it will calculate the parameters (r, θ) of that line, and then look for the accumulator's bin that the parameters fall into, and increment the value of that bin. On finding the bins with the highest values, mainly by looking in the accumulator space for local maxima, the most likely lines can be extracted, and their (approximate) geometric definitions read off. The easiest and simplest way for finding these peaks is by applying some form of threshold, but other techniques may yield better results in different circumstances - determining which lines are found as well as how many. As the lines returned do not contain any length information, it is often necessary, in the next step, to find which parts of the image match up with which lines. Due to the imperfection errors in the edge detection step, thus there will be errors in the accumulator space, which make it non-trivial to find the appropriate peaks and the appropriate lines.

The final result of the linear Hough transform is a two-dimensional array (matrix) similar to the accumulator one dimension of this matrix is the quantized angle θ and the other dimension is the quantized distance r. Each element of the matrix has a value equal to the number of points or pixels that are positioned on the line represented by quantized parameters (r, θ). So the element with the highest value indicates the straight line that is most represented in the input image [6].

a. *Hough algorithm:*

The Hough algorithm is used to locate the unique (r, θ)

coordinates of a straight line. The steps of the Hough algorithm are outlined below.

(a.) Quantize parameter space appropriately according to the image size in (*x*, *y*)-space. For Example, consider a straight line ℓ in a 100 by 100 pixel image. This line ℓ can be represented by the equation,
$r = x\cos\theta + y\sin\theta$

The values of *r* that ℓ can take lie in the range $0 \le r \le d$. Where $d = \sqrt{100^2 + 100^2}$ is the length of the diagonal across the image. Thus the values of r in parameter space are restricted by the image size in (*x*, *y*)-space. The values of θ that ℓ can take lie in the range $0 \le \theta \le 360$.

(b.) Construct an accumulator array $A(r, \theta)$ (i.e. an array with dimensions (r, θ) with all elements initially set to zero.

(c.) Examine feature points of the input image. This may involve a gradient exceeding some threshold. However this step however is optional. It is reasonable to include every point in the Hough transform. If it is not known what defines a feature points, then if there is the randomized Hough transform that is to be implemented, then a number of random points in the image will be chosen for examination.

(d.) For each (x_i, y_i) point examined, evaluate the corresponding r_i and θ_i values that the point may take. For each r_i and θ_i value obtained, add one to the $A(r_i, \theta_i)$ position in the accumulator array,

$A(r_i, \theta_i) = A(r_i, \theta_i) + 1$

This constitutes a "vote" for this r_i, θ_i value.

(e.) Examine the accumulator array for global and local maxima (i.e.(r, θ) |values that obtained significantly more votes).As these maxima correspond to collinear points.

The longest line ℓ in the input image will have the maximum value in the accumulator array; this maximum value provides a measure of how many points lie on the line. The position in the array $A(r_m, \theta_m)$where this maximum value occurs corresponds to the *r* and θ value occurs corresponds to the *r* and ɵ values that represent ℓ. Hence, ℓ can be reconstructed by the equation,

$r_m = x\cos\theta_m + y\sin\theta_m$

III. PROPOSED TECHNIQUE

The angular points of the Input finger are matched against the template finger by using the Hough transform. If the match occurs only then minutiae extraction will be used to verify the fingerprint matching using minutiae points. The performance can be analyzed by using the false matching rate.

A.Proposed algorithm:

Problem definition**:** Given the test Fingerprint Image the objectives are,

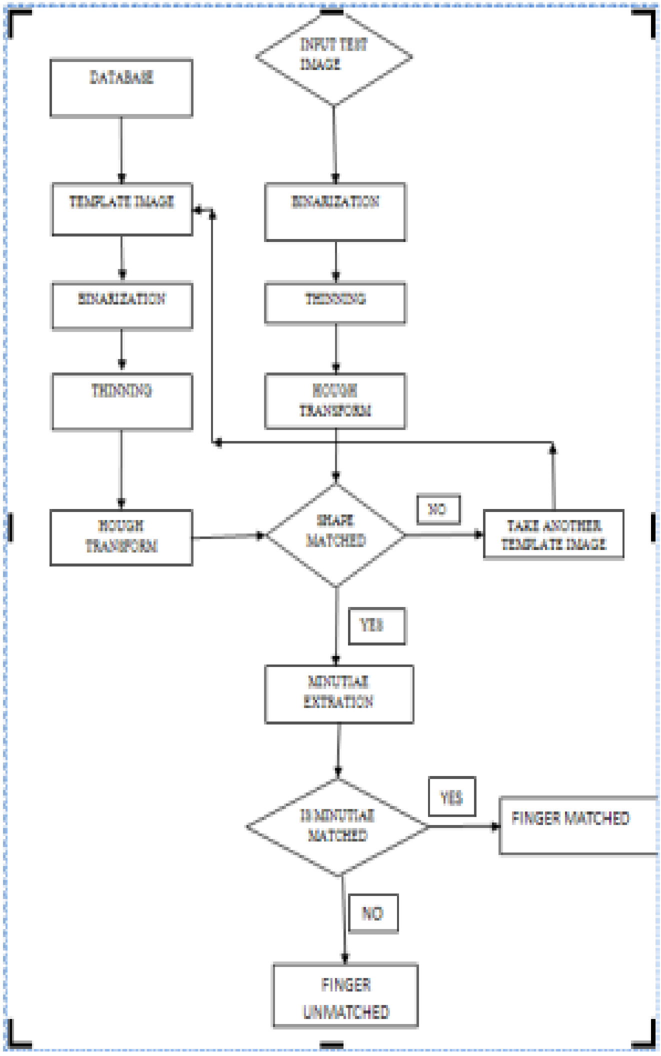
a. Pre-processing the test Fingerprint.
b. Perform the Hough transform
c. Extract the minutiae points.
d. Matching test Fingerprint with the database.

Table 1 gives the algorithm for fingerprint verification, in which input test fingerprint image is compared with template fingerprint image, for recognition. Input: Gray-scale Fingerprint image, template database having r row and c column.

Output: Verified fingerprint image with matching score.

a. Input test finger Image say img.
b. Binarized the test image (img).
c. Apply Thinning on Binarized Image (img).
d. Apply Hough transform to get the shape description of img.
e. For i=1:r
f. For j=1:c
g. Binarize the Template image(i,j) then apply thinning
h. Apply Hough transform on template image(I,j) to get shape description
i. If shape description of template image(i,j) matches shape description of img
j. Row=I;
k. Break;
l. End
m. End
n. Extract minutiae point of img say mimg
o. For j=1:c
p. Extract minutiae point of template image(row,j) say t(row,j)
q. If t(row,j)== mimg
r. Then finger matched & exit
s. End
t. If j=c+1
u. Then finger unmatched & exit.

B. Flowchart of proposed technique:



IV. MATCHING PARAMETERS

a. False Matching Ratio: when a fingerprint matches with the different fingerprint individual than it is called as false matching ratio [9].

It is given in an equation (1)

$$\mathrm{FMR} = \frac{\text{False Matches}}{\text{Im posterAttempts}} \ldots\ldots\ldots\ldots\ldots (1)$$

The attempts are implemented by matching each input image with all the template images.

The False match was recorded for each imposter attempt when the matching score was greater than the established threshold.

b. False Non Matching Ratio (FNMR)*:* when a fingerprint is not completely matches with the different fingerprint individual than it is called as False Non matching ratio. Or FNMR can be define as the probability that the system denies access to an approved user is given in an equation (2)

$$FNMR = \frac{False\ NonMatch}{EnrolleAttempts} \dots \dots (2)$$

Enrollee attempts are implemented by matching each input image with corresponding template image, it is one-to-one matching. When the matching score between an enrollee and its template was less than the established threshold's False Non-match was recorded.

b. Matching Score:

It is used to calculate the matching score between the input and template data is given in an equation (3)

$$Matchingscore = \frac{MatchingMinutiae}{Max(NT, NI)} \dots (3)$$

Where, NT and NI represent the total number of minutiae in the template and input matrices respectively. According to this, the matching score takes on a value between 0 and 1. And the Matching score of 1 and 0 indicates that data matches perfectly and data is completely mismatched respectively.

Table1: Comparison of FNMR and FMR.

	Minutiae extraction	***hough transfrom***	***Proposed Method***
FNMR	0	0	0
FMR	0.19	0.20	0.15

V. CONCLUSION AND FUTURE SCOPE

Table1 shows the Value of FNMR and FMR for the various simulation runs for the proposed scheme and the existing schemes. The false matching rate get reduced in the proposed scheme i.e. accuracy of the fingerprint matching get increased. The main limitation of the proposed scheme is the increased overhead. In future the overhead can be reduced by introducing the orientation of the finger.

VI. REFERENCES

[1]. Dr. Neeraj Bhargava, Dr.Ritu Bhargava, Prafull Narooka, MinaxiCotia "Fingerprint Recognition Using Minutia Matching" International Journal of Computer Trends and Technology- volume3Issue4- 2012.

[2]. A Tutorial on Fingerprint Recognition1 Davide Maltoni

[3]. I. Iancu, N. Constantinescu, M. Colhon, Fingerprints Identification using a Fuzzy Logic System, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. V (2010), No. 4, pp. 525-531.

[4]. Ritu "Minutiae Based Fingerprint Recognition" International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 2 Issue 4 April, 2013 Page No. 1166-1171.

[5]. Spatial Decomposition of The Hough Transform – Heather and Yang – IEEE computer Society – May 2005.

[6]. Machine Vision - Wesley E. Snyder, Hairong Qi, Cambridge University Press, 2004.

[7]. Machine Vision – Ramesh Jain, Rangachar Kasturi, Brian G Schunck, McGraw-Hill, 1995.

[8]. Ravi. J, K. B. Raja, Venugopal. K. R "Fingerprint Recognition Using Minutia Score Matching" International Journal of Engineering Science and Technology Vol.1 (2), 2009, 35-42 35

[9]. Hemlata Patel#1, Mr. Vishal Sharma "Fingerprint Recognition by Minutiae Matching Method for Evaluating Accuracy "International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue5- May 2013 ,ISSN: 2231-5381 http://www.ijettjournal.org Page 2136

[10]. Kalyani Mali, Samayita Bhattacharya, "Fingerprint Recognition Using Global and Local Structures", International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397 Vol. 3 No. 1 Jan 2011, 16.

(www.ijarcs.info))