

Extended Feature Extraction Technique From Fingerprint

Satyabrata Swain
Dept. of Computer Science
NIT Rourkela
Rourkela 769008, India
Email: satya.swain10@gmail.com

Banshidhar Majhi
Dept. of Computer Science
NIT Rourkela
Rourkela 769008, India
Email:bmajhi@nitrkl.ac.in

Ratnakar Dash
Dept. of Computer Science
NIT Rourkela
Rourkela 769008, India
Email:ratnakar@nitrkl.ac.in

Abstract—Automated Fingerprint Identification Systems (AFIS) currently rely only on Level 1 and Level 2 features. But these features are not much helpful for forensic experts as the experiment deals with partial to full print matching of latent fingerprint. Forensic experts takes the advantage of extended feature proposed by “Committee to Define an Extended Fingerprint Feature Set” (CDEFFS). This paper presents extraction technique of two extended features, dots and incipient ridges by tracing valleys. We have found starting points on the valley by analyzing the frequencies present in the fingerprint. Valley are traced from the starting point using first marching method (FMM). Then an intensity checking method is used for finding these features. Extensive simulation is carried out in MATLAB environment to show the superiority of the proposed feature extraction technique over state of art. Accuracy of the proposed feature extraction scheme also has been shown using special database 30 and IIT Delhi database.

Index Terms—fingerprint, extended feature, level 3 feature, dots and incipient

I. INTRODUCTION

A number of biometric traits have been developed and several of them are being used in variety of applications. Among these, fingerprint, face, iris, speech and hand geometry are most commonly used system. Fingerprint recognition has been accepted as a reliable person identification technique for almost hundred years due to their uniqueness and immutability. Although fingerprint verification is usually associated with criminal identification and police work, now-a-days it has become more popular in civilian and commercial applications, such as access control, financial security and verification of firearm purchasers, etc.

Skin on human fingertips, called as fingerprint, contains ridges and valleys (furrows) which together forms distinctive patterns. These patterns are fully developed in the womb and are permanent throughout whole lifetime. Fingerprint is having rich set of feature. These features are generally categorized into three levels. Level 1 features are defined by fingerprint ridge flow (ridge pattern types, ridge orientation field, and singular points). Level 2 features, also called as Galton characteristics or minutiae, such as ridge bifurcations and ridge endings. Level 3 features are defined as fingerprint ridge dimensional features. Pores, dots, incipient ridges, ridge edge shapes, breaks, creases, scars, and other permanent details are defined as level 3 (named as extended feature by CDEFFS)

features.

Level 1 features are not unique, so it is mainly used for type classification (e.g. whorl, left loop, right loop, and arch). Level 2 features are quite discriminative and stable, so successfully used in AFIS. In case of latent print examination (full to partial matching), level 2 features are not sufficient as the area of the fingerprint surface is very small (so contains less number of minutiae). According to the forensic experts Level 3 features are permanent, immutable, unique, and also provide discriminatory information for human identification. However, level 3 feature can be captured using high resolution sensor (≥ 1000 pixels per inch (ppi)) where as level 2 can be captured using 500 ppi sensor. Dots and incipient ridge are present in valley in between two ridge. Its size is less than 0.02 inch, where as incipient ridge is thinner than local ridge width. A lots of work has been reported on level 3 feature. Anil K. Jain *et al.* [2] proposed a method for extracting pores and ridges using wavelet transform and Gabor filters. N. Manivanan *et al.* [3] proposed pore extraction method using highpass and correlation filtering. Yi Chen *et al.* [1] proposed a method for dots and incipient ridge extraction by using local phase symmetry and continuous skeletonization of valley, this skeletonization process is computationally expensive. In this paper we have proposed a technique for the extraction of dots and incipient ridges by tracing valleys. Rest of the paper is organized as follows: Section II explains extraction of starting point. Section III gives description on FMM. Section IV introduces proposed extraction technique. Finally experimental result and, conclusion are presented in section V and VI respectively.

II. STARTING POINT EXTRACTION

Our primary objective is to select a set of points on valleys from which tracing of valley can be started. Tracing is done in such a way that each traced point is approximately on the middle of that valley. Tracing needs a location of a starting point. Better starting point provides a better tracing. A window is taken from the middle portion of the fingerprint. X-signature [5] of the window is calculated for analyzing the frequencies present in the window. This window consists of signals with different frequencies. Ridge valley pattern of a fingerprint image can be represented as a sinusoidal wave with constant

frequency. In signal processing concept if the speed of the signal is equal to wavelength, then the frequency component contains only first harmonic. The frequency is calculated as,

$$Frequency = \frac{Speed}{Wavelength} = \frac{\lambda_r + \lambda_v}{\lambda} = 1 \quad (1)$$

where λ_r and λ_v is the width of the ridge and valley respectively. The above frequency represents the first harmonics.

To extract the ideal signal, all the frequencies above and below the first harmonics are suppressed. The lower frequency component is due to the presences of noise along with dots and incipient ridge on the valley and the high frequency component is due to noise along with pores on the ridges. The ideal signal is obtained by window based filtering with its corresponding power spectrum. The processed X-signature Fig. 1(d) is approximately close to a signal which contains only single frequency. The peak of the processed X-signature will be the center of the valley. As we need a starting point on each of the valley on a fingerprint, window is taken from the middle of the image and vertically moved down. The reason for selecting the window at the middle of the image is to cover the maximum number of valleys with minimum number of windows. The problem may arises near singular point and delta region. In order to get a starting point in these regions, we shift the window towards left and right horizontally until a starting point is identified.

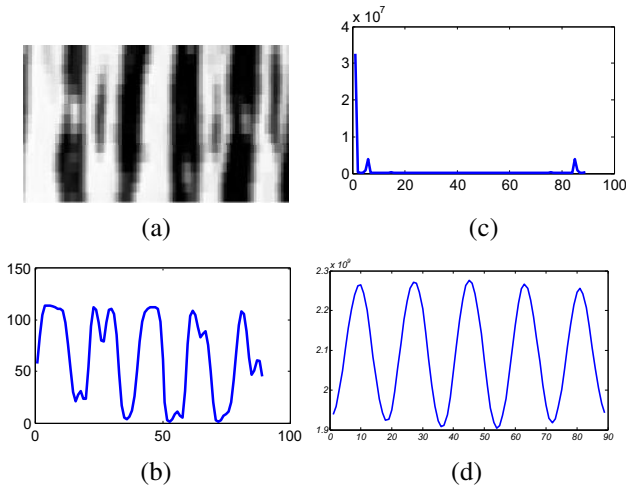


Fig. 1. (a) window from the middle part of fingerprint, (b) X-Signature of the window, (c) Power spectrum of the X-Signature, (d) Processed X-Signature

The location of the starting point on the valley is ensured by the above mentioned technique Fig. 1. However, the point may located either on white area or on black area (dots or incipient ridge). To know that, an intensity checking procedure is used. The intensity of the point is compared with a threshold (experimentally calculated) and if it is less than threshold then it is discarded as starting point but can be further processed for dots or incipient ridges detection. If intensity is above the threshold then entropy of the neighborhood of the point is calculated using (2). Entropy based checking is done to know

the presence of white noise on the dots and incipient ridge. If the starting point is on the white patch (noise) in the dots then entropy of the neighborhood will be higher. So if entropy of the neighborhood is higher than a threshold value then the point is discarded otherwise it is considered as starting point.

$$Entropy = - \sum_{i=0}^{i=255} P_j \log_2 P_j \quad (2)$$

where P_j is the probability that the difference between two adjacent pixel is i . After finding the starting point, the next job is to trace the valley using FMM. FMM [9] is utilized for image segmentation. In this paper, we have used FMM for tracing valleys. Following section explains the concept of FMM.

III. FIRST MARCHING METHOD

The fast marching method is a numerical method for solving boundary value problems of the Eikonal equation. It is a special case of level set method for computing the propagation of front where the speed depends on the local position and is extremely fast scheme for solving the Eikonal equation. We have explained the whole concept in following steps:

A. Interface Propagation

Given an arbitrary region, the interface is defined as the curve, or surface, separating the area inside of the region from the area outside of the region. Interface propagation is defined as the motion of the interface in normal direction. For motion, a velocity is required which is described as 1) a component depends on the local property of the interface, 2) a component belonging to global properties of the surface, 3) and a component independent of the surface.

B. The Boundary Value Formulation

An interface which is strictly expanding or contracting is known as the boundary value formulation. Eikonal equation establish a relation between arrival function and speed of the interface as follows.

$$|\nabla T| F = 1 \quad (3)$$

where T is the arrival function and F is the speed of the interface. Arrival function has one more dimension than the surface that is if surface is in R^2 then the arrival function will be in R^3 . For the boundary value formulation, interface is expanding or contracting if $F > 0$ and $F < 0$ respectively. Fast Marching Method is the special case where the interface move in one direction, where as the initial value formulation does not impose this restriction.

C. Initial Value Formulation

In the initial value formulation, an interface may propagate back to points it has already propagated. It follows that the speed function F is no longer strictly greater or strictly less than 0 for all time. In order to facilitate this more general definition, a level set function is required.

D. Level set method and its solution

Given an initial position for an interface τ , where τ is a closed curve in R^2 , and a speed function F , which gives the speed of τ in its normal direction, the level set method takes the perspective of viewing τ as the zero level set of a function $\Phi(x, t = 0)$ from R^3 to R . That is $\Phi(x, t = 0) = \pm d$, where d is a distance from x to τ , and the plus (minus) sign is chosen if the point x is outside (inside) the initial surface τ . Using chain rule, an evolution equation for the interface can be written as,

$$\Phi_t + |\nabla\Phi| F = 0 \quad (4)$$

where Φ is the level set function, F is the speed function, and Φ_t represents the first derivative of Φ with respect to t . Solution of above equation is given by,

$$\begin{aligned} \Phi_{ij}^{n+1} = & \Phi_{ij}^n - \Delta t (\min(D_{ij}^{+x}\phi, 0)^2 \max(D_{ij}^{-y}\Phi, 0))^{1/2} \\ & - \Delta t (\max(D_{ij}^{-x}\Phi, 0)^2 + \min(D_{ij}^{+y}\Phi, 0)^2)^{1/2} \end{aligned} \quad (5)$$

where the speed is $F = 1$ and the difference operator is defined as $D_{ij}^{+x}\Phi = (\Phi_{i+1,j} - \Phi_{i,j})/\Delta t$. This solution is for evolution of all the level sets corresponding to the front itself. So this is computationally expensive solution. An efficient technique called ‘‘narrow band approach’’ which works only in a neighborhood of the zero level set. Using this concept, grid points are divided into alive, land mines or far away, depending on whether they are inside the band, near its boundary, or outside the band, respectively. One - cell version of this leads to fast marching method.

E. Fast Marching Level Set Method

If $F = F(x, y, z) > 0$ is a speed function, a monotonically advancing front whose level set equation is $\Phi_t + F(x, y, z)|\nabla\Phi| = 0$ is given by,

$$\Phi(x, t = 0) = \tau \quad (6)$$

In two-dimensional case in which the interface is a propagating curve, zero level set is evolving above xy-plane. $T(x, y)$ be the time at which the curve crosses the point (x,y). The surface will satisfy Eikonal equation (3).

According to (3), gradient of arrival time surface is inversely proportional to the speed of the front. Using approximation of the gradient, the solution of the above equation is given as,

$$\begin{aligned} \max(D_{ij}^{-x}T, 0)^2 + \min(D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, 0)^2 + \\ \min(D_{ij}^{+y}T, 0)^2 = \frac{1}{F^2} \end{aligned} \quad (7)$$

where $T(x, 0) = 0$.

Rouy and Tourin find a less diffusive iterative algorithm for computing the solution for Eikonal equation which is given as,

$$\begin{aligned} \max(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T, 0))^2 + \\ \max(\max(D_{ij}^{-y}T, 0), -\min(D_{ij}^{+y}T, 0))^2 = \frac{1}{F_{ij}^2} \end{aligned} \quad (8)$$

The important aspect of constructing Fast Marching Level Set Method algorithm is to estimate propagation of front in upwind manner with speed F_{ij} and its direction is normal to each grid point. Here the set of grid points $j = 1$ correspond to y axis, and we assume that $F_{ij} > 0$. So the algorithm solves the above eq by increasing the value of T from its smallest value. Front will sweep in upwind direction by considering a set of points in the band around the existing front, freezing the values of existing points and bringing new ones into the narrow band structure. The next step is to select the grid point to be updated within the current narrow band. The algorithm [9] is stated in following steps

- 1) Initialize
 - a) (Alive points: shaded points): Let A be the set of all grid points $i, j = 1$; set $T_{i,1} = 0.0$ for all points in A .
 - b) (Narrow band points: circles): Let Narrow Band be the set of all grid points $i, j = 2$; set $T_{i,1} = \frac{dy}{F_{ij}}$ for all points in Narrow Band.
 - c) (Far away points: rectangles): Let Far Away be the set of all grid points $i, j > 2$; set $T_{i,j} = \infty$ for all points in Far Away.
- 2) Marching Forward
 - a) Begin Loop: Let (i_{\min}, j_{\min}) be the point in Narrow Band with the smallest value for T .
 - b) Add the point (i_{\min}, j_{\min}) to A ; remove it from NarrowBand.
 - c) Tag as neighbors any points $(i_{\min} - 1, j_{\min}), (i_{\min} + 1, j_{\min}), (i_{\min}, j_{\min} - 1), (i_{\min}, j_{\min} + 1)$ that are either in Narrow Band or Far Away. If the neighbor is in Far Away, remove it from that list and add it to the set Narrow Band.
 - d) Recompute the values of T at all neighbors according to Equation 8, selecting the largest possible solution to the quadratic equation.
 - e) Return to loop.

The important aspect of constructing Fast Marching Level Set Method algorithm is to estimate propagation of front in up- wind manner with speed F_{ij} and its direction is normal to each grid point.

IV. FEATURE EXTRACTION AND MATCHING

We have used FMM for tracing valleys of the fingerprint. Instead of taking initial point (alive point as defined in FMM) along y-axis as FMM, the set of points are taken as extracted in section II. We have able to propagate toward right and left of the saddle point which are normal to the interface in anti-clockwise and clockwise direction respectively. While tracing, we are taking a window around the traced point. If $I(i, j) < \alpha$, where $I(i, j)$ is any pixel in the window, and α is a gray level intensity value, then a connected component is constructed by considering the range of the intensity value 0 to α . These connected component includes dots, incipient ridge and noise. CDEFFS standardized that the size of dots and

incipient ridges are less than 0.02 inch. So all the extracted components whose size is more than 0.02 inch (part of a ridge) are discarded. Centroid of rest of the extracted component and its direction are calculated. The fingerprint with traced valley and the extracted dots and incipient are shown in Fig 2 (a) and 2 (b) respectively. We have extracted minutiae by using a commercial software called MINDCT of NIST Biometric Image Software (NBIS) [11]. The extracted centroid of dots and incipient ridge are added with the minutiae to form a new set of feature vector. Feature vectors are input to BOZORTH3 of NBIS for calculating matching score.

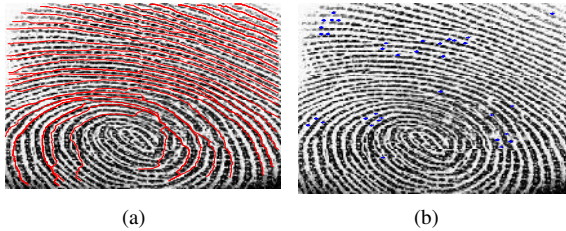


Fig. 2. (a) Traced image, (b) Dot and incipient are shown in '*' and '+' respectively

V. EXPERIMENTAL RESULTS

In this paper we have matched between partial to full fingerprint using dots and incipient ridge. We have used NIST special database 30 (SD30) [12] and Rural Indian Fingerprint Database [7] of IIIT Delhi for performance evaluation. All images in SD30 are taken in constrained environment and images in IIIT database are taken in unconstrained environment. The earlier database includes 72 tenprint cards from 36 users, 10 fingers per user and 2 impressions per finger, scanned at both 500ppi and 1000ppi. The latter includes each 150 fingerprint of right hand and left hand index fingerprint image of urban and rural people and 10 print per each individual. So the total number of print is $3000((75 \times 2 \times 10) + (75 \times 2 \times 10))$. For performance measure, two experiments have been conducted for each database. In the first experiment on SD30, the first impression of each fingerprint is divided into 16 non-overlapping block of size 350×350 and rest pixels at boundary are discarded. The total number of genuine and impostor match are 5760 and 2067840 respectively.

Similarly, in case of Rural Indian Fingerprint Database, the first image is divided into 12 non-overlapping block of size 150×150 and rest pixels are discarded at boundary. Number of probe image is 3600 (300×12). Total number of genuine and impostor match are 3600 and 1076400 respectively. In the above case, we are taking non-over lapped partial print for matching against full print assuming that the matching is a partial to full matching. However, above case is not true in case of live data. For that we have randomly cropped 10 partial prints from the first impression and match them against second full impression. The total number of genuine and impostor scores are 3600 and 1292400 in case of NIST database, and 3000 and 897000 number for IIIT Delhi Fingerprint database respectively. Fig. 3 and Fig. 4 shows the acceptance rate of

our approach with SD30 and IIITD database respectively. The result has been shown using only level 2 feature and level 2 along with level 3 feature. Accuracy of proposed method has been compared with Yi Chen [1] method in Table I. In [1], they have cropped the partial print manually but in our case we have divided a full print into non-overlapped partial print. This method ensures that a partial print contain a region from all parts of a full fingerprint. Our method shows superiority over Yi Chen [1] method.

VI. CONCLUSION

In this paper, a novel approach for extracting two extended features, dots and incipient ridges from fingerprint have been proposed. Partial to full print matching by taking non-overlapped block and random block which shows the effectiveness of the proposed approach over the state of art method. The proposed extraction technique would be benefit not only for forensic expert but may be used in Next Generation Identification Technology (NGI). A new matching technique with the proposed feature can give a better classification accuracy.

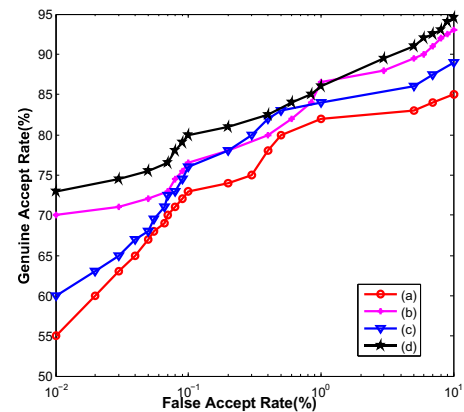


Fig. 3. ROC using SD 30 database (a) non-overlapped cropping using only level 2 features, (b) non-overlapped cropping using level 2 and level 3 features (c) random cropping using only level 2 features (d) random cropping using level 2 and level 3 features

REFERENCES

- [1] Chen, Yi, and Anil K. Jain. "Dots and incipients: extended features for partial fingerprint matching." *In Biometrics Symposium*. IEEE 2007, pp. 1-6.
- [2] Jain, Anil, Yi Chen, and Meltem Demirkus. "Pores and ridges: Fingerprint matching using level 3 features." *ICPR 2006, 18th International Conference on Pattern Recognition*. IEEE, 2006. pp. 477-480.
- [3] Manivanan, N., S. Memon, and W. Balachandran. "Automatic detection of active sweat pores of fingerprint using highpass and correlation filtering." *Electronics letters* 46 vol 18 pp. 1268-1269, 2010.
- [4] Zhao, Qijun, Jianjiang Feng, and Anil K. Jain. Latent fingerprint matching: Utility of level 3 features. *MSU Technical Report* 8. pp. 1-30, 2010.
- [5] Arpit, Devansh, and Anoop Nambodiri. "Fingerprint feature extraction from gray scale images by ridge tracing". *In Biometrics (IJCB), 2011 International Joint Conference on*. 2011, IEEE, pp. 1-8.
- [6] "CDEFFS, ANSI/NIST Committee to Define an Extended Fingerprint Feature Set" <http://fingerprint.nist.gov/standard/cdeffs/index.html/>.

TABLE I
 ACCURACY COMPARISON OF PROPOSED APPROACH WITH YI CHEN'S APPROACH FOR LEVEL 2 AND LEVEL 2 WITH LEVEL 3 FEATURES USING SD 30
 AND IIIT DELHI DATABASE.

Database	Proposed Approach				Chen Approach [1]			
	Non-Overlapped		Random Cropping		Manual Cropping		Random Cropping	
	level 2 Feature	Level 2 + 3 Feature	Level 2 Feature	Level 2+3 Feature	Level 2 Feature	Level 2 +3 Feature	Level 2 Feature	Level 2 +3 Feature
NIST 30	82	88	85	90	82	87	85	90
IIIT Delhi	79	86	82	86	79	85	82	85

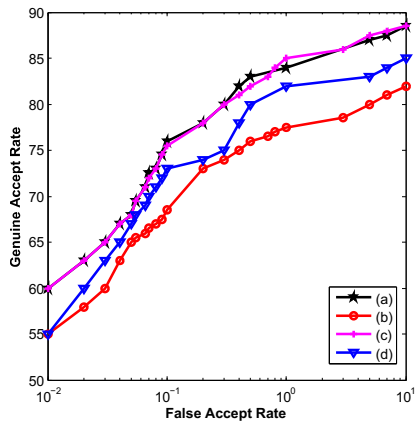


Fig. 4. ROC using IIIT Delhi database (a) random cropping using level 2 and level 3 features (b) non-overlapped cropping using only level 2 features (c) non-overlapped cropping using level 2 and level 3 features (d) random cropping using only level 2 features

[7] Puri, C., K. Narang, A. Tiwari, M. Vatsa, and R. Singh. "On analysis of rural and urban indian fingerprint images". In *Ethics and Policy of Biometrics*. Springer Berlin Heidelberg, 2010 pp. 55-61.

[8] Jain, Anil K., Yi Chen, and Meltem Demirkus. "Pores and ridges: High-resolution fingerprint matching using level 3 features". *Pattern Analysis and Machine Intelligence*. IEEE Transactions on 2, vol. pp. pp. 115-27.

[9] Sethian, James A. "A fast marching level set method for monotonically advancing fronts" *Proceedings of the National Academy of Sciences* 93, vol. 4 pp. 1591-1595, 1996.

[10] Dicker, Jeff. "Fast marching methods and level set methods: an implementation". *PhD diss.*, University of British Columbia, 2006.

[11] "NIST Biometric Image Software (NBIS) 4.2.0", <http://www.nist.gov/itl/iad/ig/nbis.cfm>

[12] "National Institute of Standards and Technology, "NIST Special Database 30," <http://www.nist.gov/srd/nist30.htm>.