Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Fingerprint Classification Based on Maximum Variation in Local Orientation Field

Shekhar Suralkar
Electronics dept.
SSBTE, Bambhori
Jalgaon , India

Milind E Rane, Pradeep M. Patil
Electronics dept.
Vishwakarma Institute of technology,
Pune, India
Patil_pm@rediffmail.com

Abstract— **Fingerprint classification provides an important indexing mechanism in a fingerprint database. Accurate and consistent classification can greatly reduce fingerprint-matching time and computational complexity for a large database as the input fingerprint needs to be matched only with a subset of the fingerprint database. Classification into six major categories (whorl, right loop, left loop, twin loop, arch, and tented arch) with no reject options yields an accuracy of 89.7 %. The overall accuracy is improved to 91.5 % if the arch and tented arch are merged as a single class. The penetration rate of the proposed classification system is 88.9%.**

Keywords— Fingerprint classification, whorl, right loop, left loop, twin loop, arch, orientation field, singularity points

I. INTRODUCTION

The identification of a person requires a comparison of his fingerprint with all the fingerprints in a database. This database may be very large (e.g., several million fingerprints) as in many forensic and civilian applications. In such cases, the identification typically has an unacceptably long response time. The identification process can be speeded up by reducing the number of comparisons that are required to be performed. A common strategy to achieve this is to divide the fingerprint database into a number of bins (based on some predefined classes). A fingerprint to be identified is then required to be compared only to the fingerprints in a single bin of the database based on its class. Fingerprint classification refers to the problem of assigning a fingerprint to a class in a consistent and reliable manner. Although fingerprint matching is usually performed according to local features (e.g., minutiae), fingerprint classification is generally based on global features, such as global ridge structure and singularities.

Fingerprint classification is a difficult pattern recognition problem due to the small inter-class variability and the large intra-class variability in the fingerprint patterns Moreover, fingerprint images often contain noise, which makes the classification task even more difficult. The selectivity of classification-based techniques strongly depends on the number of classes and the natural distribution of fingerprints in these classes. Most of the existing fingerprint classification methods can be coarsely assigned to one of these categories: rule-based, syntactic, structural, statistical, neural network-based and multi-classifier approaches.

A fingerprint can be simply classified according to the number and the position of the Singularities. In [1], the Poincare index is exploited to find type and position of the singular points and a coarse classification is derived. The problem with this method is that fingerprints of the Arch type do not have any singularity in terms of Pointcare Index so structural heuristic is used to locate the core point. A syntactic method describes patterns by means of terminal symbols and production rules; a grammar is defined for each class and a parsing process is responsible for classifying each new pattern [2, 3]. Structural approaches are based on the relational organization of low-level features into higher-level structures. This relational organization is represented by means of symbolic data structures, such as trees and graphs, which allow a hierarchical organization of the information [4]. These regions and the relations among them contain information useful for classification. In statistical approaches, a fixed-size numerical feature vector is derived from each fingerprint and a general-purpose statistical classifier is used for the classification [5]. Many approaches directly use the orientation image as a feature vector, by simply nesting its rows [6, 7]. Most of the proposed neural network approaches are based on multilayer perceptrons and use the elements of the orientation image as input features [8]. A pyramidal architecture [9,10,11] constituted of several multilayer perceptrons, each of which is trained to recognize fingerprints belonging to a different class. Neural networks can be extensively used for fingerprint classification [12]. Fingerprints are classified as Lasso or Wirbel [13].

In this paper we introduce a process for classification of fingerprints based on maximum variation in local orientation field. The paper has been organized as follows. Section II introduces the proposed algorithm. Section III explains the required parameters for judging the performance of fingerprint classification. Section IV displays the results and a brief discussion is carried out based on these results.

II. PROPOSED ALGORITHM

The proposed algorithm is divided in two steps. The first step computes the singularity points on the fingerprint image based on the maximum variation of its local orientation. The second step classifies the fingerprint based on the location of the detected core and delta points.

Step I Singular Point Detection through vector field computation:

1) Divide the input image, into non-overlapping blocks of size 8×8 pixels.

2) Compute the gradients $\partial_x(i,j)$ and $\partial_y(i,j)$ at each pixel (i,j). Depending on the computational requirement, the gradient operator may vary from the simple *Sobel* operator to the more complex *Marr-Hildreth* operator.

3) Estimate the local orientation of each block centered at pixel (i,j) using

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

where,

$$V_x(i,j)=\sum_{u=i-4}^{i+4}\sum_{v=j-4}^{j+4}2\partial_x(u,v)\partial_y(u,v) \tag{2}$$

$$V_y(i,j)=\sum_{u=i-4}^{i+4}\sum_{v=j-4}^{j+4}\left(\partial_x^2(u,v)-\partial_y^2(u,v)\right) \tag{3}$$

The value of $o(i,j)$ is least square estimate of the local ridge orientation in the block centered at pixel (i,j). Mathematically, it represents the direction that is orthogonal to the dominant direction of the Fourier spectrum of the 8×8 window.

4) Convert the orientation field in to range of 0 to 180 degree.

$$o(i,j)=\begin{cases} o(i,j) & \text{if } o(i,j)<\pi \\ \pi+o(i,j) & \text{if } o(i,j)\le -\pi/2 \\ o(i,j)-\pi & \text{otherwise} \end{cases} \tag{4}$$

5) Smooth the orientation field in a local neighborhood. In order to perform smoothing (low pass filtering), the orientation image needs to be converted into a *continuous vector field*, which is defined as

$$\phi_{1x}(i,j)=\cos\left(2\,o(i,j)\right) \tag{5}$$

and

$$\phi_{1y}(i,j)=\sin\left(2\,o(i,j)\right) \tag{6}$$

where, ϕ_{1x} and ϕ_{1y}, are the x and y components of the vector field, respectively. With the resulting vector field, the low pass filtering can be performed as

$$\phi_x(i,j)=\sum_{u=-w/2}^{w/2}\sum_{v=-w/2}^{w/2}W(u,v)\phi_{1x}(i-wu,j-wv) \tag{7}$$

and

$$\phi_y(i,j)=\sum_{u=-w/2}^{w/2}\sum_{v=-w/2}^{w/2}W(u,v)\phi_{1y}(i-wu,j-wv) \tag{8}$$

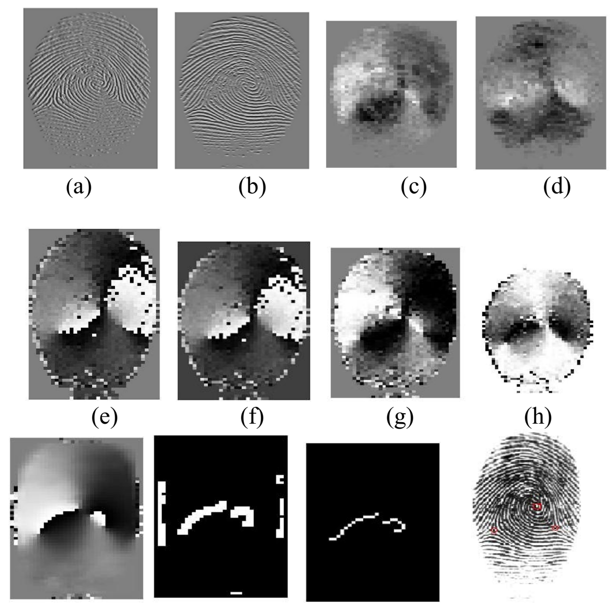
where $W(.)$ is a two dimensional low pass filter with unit integral and $w\times w$ specifies the filter size. Note that smoothing operation is performed at the block level. For our experimentation we have used a 5×5 mean filter. The smoothed orientation field O at (i,j) is computed as

$$O(i,j)=\frac{1}{2}\tan^{-1}\left(\frac{\phi_y(i,j)}{\phi_x(i,j)}\right) \tag{9}$$

6) Extract the part of image having maximum variation in intensity using mask of size 3×3.

7) Thin the image.

8) The singularity detected is referred as delta if the pixel below the singularity point is having angle less than $60°$ in the orientation field, otherwise it is referred as core.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig 1 Various steps explaining the proposed algorithm (a)gradient $\partial_x(i,j)$ (b)gradient $\partial_y(i,j)$ (c) $V_x(i,j)$ (d) $V_y(i,j)$ (e)Local orientation field $0°-360°$ (f) Local orientation field $0°-180°$ (g) x component of continuous vector field (h) y component of continuous vector field (i) smoothened orientation field (j) extracted area of max variation in local orientation (k) Thinning of maximum variation in local orientation image (l) Detected singularity points.

Fig 1 shows the complete process of obtaining the singularity points based on the local field orientation. Fig 1(a) and (b) are obtained by computing the gradient $\partial_x(i,j)$ and gradient $\partial_y(i,j)$ respectively. The local orientation for each block is calculated using (1) to obtain Fig 1(e). The Local orientation field $0°-360°$ obtained in Fig 1(e) is converted to $0°-180°$ using (4) as shown in Fig 1(f). The orientation

image is converted into a *continuous vector field* by (5) and (6) to obtain x component of continuous vector field as shown in Fig 1(g) and y component of continuous vector field as shown in Fig 1(h). Smoothing of image is carried out using (9) to obtain Fig 1(i). The part of image having maximum variation of intensities is extracted using mask of size 3×3 to obtain Fig 1(j). This figure is then thinned using thinning algorithm to obtain a line joining the singularity points as shown in Fig 1(k). The detected singularity points can be seen in Fig. 1(l).
The proposed algorithm is implemented on various classes of fingerprints. Fig 2 shows each class of fingerprint along with their detected singularity points.

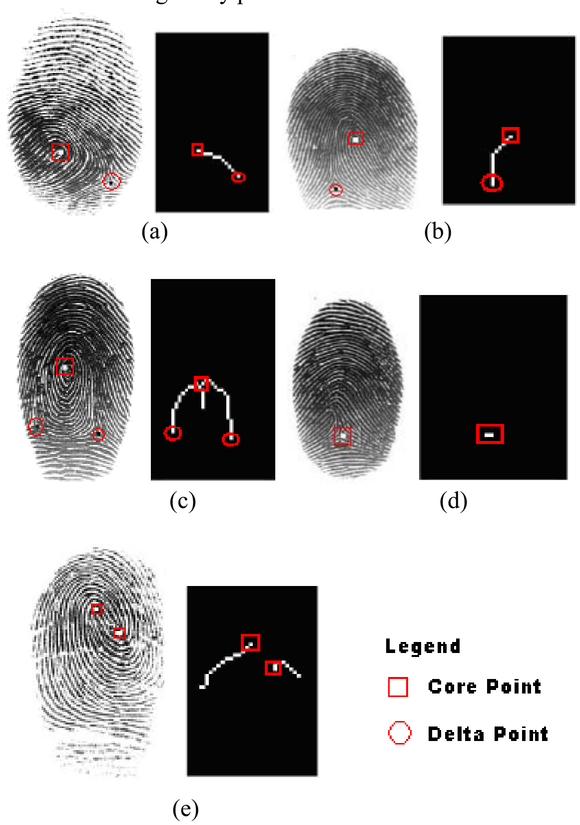


Fig 2 Classification of fingerprints using singularity points obtained through maximum variation of orientation field
(a) Left loop (b) Right loop (c) Whorl (d) Arch (e) Twin loop

Step II Classification of fingerprints based on singular points:
After obtaining the thinned image from the maximum variation of local orientation field as displayed in Fig 1(k) it can be classified into left loop, right loop, twin loop, whorl and arch by adapting the following procedure
a) If the fingerprint has zero delta points or zero core points, that is it has only one singularity point then the fingerprint is an arch as shown in Fig 2(d), else if one core and one delta are aligned vertically then the fingerprint is a tented arch.
b) Else If the fingerprint has exactly one delta point and one core point, and if the core point lies to the left of the delta point then the fingerprint is a left loop as shown in Fig 2(a), else the core point lies to the right of the delta point and hence it is a right loop.
c) Else If a bifurcation is obtained in the image and if delta points are obtained on either side of the bifurcation point then it is to be classified as a whorl as shown in Fig 2(c). If an image consists of two separate core points along with two deltas then the fingerprint is a special class of whorl.
d) Else If two core points are obtained in the image then the image is to be classified as a twin loop.

III. PERFORMANCE OF FINGERPRINT CLASSIFICATION

The performance of a fingerprint classification system is usually measured in terms of error rate or accuracy. The error rate is computed as the ratio between the number of misclassified fingerprints and the total number of samples in the test set; the accuracy is simply the percentage of correctly classified fingerprints:

$$error\ rate = \frac{number\ of\ misclassified\ fingerprints \times 100}{total\ number\ of\ fingerprints}\ \% \quad (10)$$

$$accuracy = 100\,\% - error\ rate \quad (11)$$

The error rate of a classification system is generally reported as a function of the percentage of the database that the system has to search; this percentage is called penetration rate and can be computed as

$$penetration\ rate = \frac{number\ of\ accepted\ fingerprints \times 100}{total\ number\ of\ fingerprints\ in\ the\ database}\ \% \quad (12)$$

A more detailed analysis of the behavior of a classifier can be obtained by examining the confusion matrix. This matrix has a row for each true class and a column for each hypothesized class; each cell at row r and column c reports how many fingerprints belonging to class r are (in)correctly assigned to class c.

IV. RESULTS AND DISCUSSION

The algorithm was executed using MATLAB version 6.3 on a P-IV, 1.2 GHz computer. Randomly selected 640 fingerprint images from the FVC 2000 database were used for classification. As per the FVC-2000 specifications the fingerprints were acquired by using a low cost capacitive sensor. The maximum rotation of acquired fingerprints is in the range of $\pm 15°$.
The over all classification accuracy of the algorithm with no reject option is found to be 91.5%. Accuracy for all the classes can be observed from the confusion matrix shown in Table 1. It can be observed that Left loop and Right loop are more confused with Arch. This confusion may occur because in Tented Arch the core and delta points are vertically aligned. In case the delta is slightly misaligned the algorithm may consider it to be a Right or Left loop. In Twin Loop 100% accuracy is attributed to the fingerprint having two distinct core points and also because the database has comparatively

Table 1 Confusion matrix for FVC-2000 test set

True Class	Assigned Class					Total	Accuracy %
	Left Loop	Right Loop	Whorl	Arch	Twin Loop		
Left Loop	146	0	2	20	0	168	86.90
Right Loop	0	196	4	16	0	216	90.74
Whorl	9	10	167	0	6	192	86.97
Arch	2	0	0	52	2	56	92.85
Twin Loop	0	0	0	0	8	8	100

less number of such fingerprints. The average time taken for executing the algorithm for detection of singularity points is 2.39 seconds and classification of all the fingerprints into 5 classes takes 4.25 seconds. As the singularity points are being extracted form the orientation field it makes the output highly noise tolerant. The penetration rate of the proposed classification system is 88.9%.

V. CONCLUSIONS

A novel approach towards fingerprint classification based on singularity points has been proposed. Accurate and consistent classification greatly reduces fingerprint-matching time and computational complexity for a large database. It is a sophisticated method for detection of singularity points and is highly noise tolerant. The core and delta points are obtained accurately from the maximum variation in the local orientation field.

REFERENCES

[1] Kawagoe M. and Tojo A., "Fingerprint Pattern Classification," Pattern Recognition, vol. 17, pp. 295-303,1984.

[2] Fu K.S. and Booth T.L., "Grammatical Inference: Introduction and Survey: Part I," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 3, pp. 343-360, 1986.

[3] Fu K.S. and Booth T.L., "Grammatical Inference: Introduction and Survey: Part II," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 3, pp. 360-376, 1986.

[4] Bunke H., "Structural and Syntactic Pattern Recognition," Handbook of Pattern Recognition and Computer Vision, C.H. Chen et al. (Eds.), World Scientific, River Edge, NJ, 1993.

[5] Jain A.K., Duin P.W., and Mao J., "Statistical Pattern Recognition: a Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

[6] Cappelli R., Maio D., and Maltoni D., "Fingerprint Classification Based on Multi-space KL," in Proc. Workshop on Automatic Identification Advanced Technologies, pp. 117-120,1999.

[7] Candela G.T., Grother P.J., Watson C.I., Wilkinson R.A., and Wilson C.L., "PCASYS — A Pattern-Level Classification Automation System for Fingerprints," Tech. Report: NIST TR 5647, Aug. 1995.

[8] Hughes P. and Green A., "The Use of Neural Networks for Fingerprint Classification," in Proc. 2nd Int. Conf. on Neural Networks, pp. 79-81,1991.

[9] Kamijo M., Mieno H., and Kojima K., "Classification of Fingerprint Images Using a Neural Network," Systems and Computers in Japan, vol. 23, pp. 89-101,1992.

[10] Kamijo M., "Classifying Fingerprint Images Using Neural Network: Deriving the Classification State," in Proc. Int. Conf. on Neural Networks, vol. 3, pp. 1932-1937,1993.

[11] Pal S.K. and Mitra S., "Noisy Fingerprint Classification Using Multilayer Perceptron with Fuzzy Geometrical and Textural Features," Fuzzy Sets and Systems, vol. 80, no. 2, pp. 121-132, 1996.

[12] Jain A.K., Prabhakar S., and Hong L., "A Multichannel Approach to Fingerprint Classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 348-359,1999.

[13] Karu K., and Jain A.K., "Fingerprint Classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.