An Iterative Fingerprint Enhancement Algorithm Based on Accurate Determination of Orientation Flow

Simant Dube*
Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur 208016, India
Email: simantd (at) iitk.ac.in

July 2, 2009

Abstract

We describe an algorithm to enhance and binarize a fingerprint image. The algorithm is based on accurate determination of orientation flow of the ridges of the fingerprint image by computing variance of the neighborhood pixels around a pixel in different directions. We show that an iterative algorithm which captures the mutual interdependence of orientation flow computation, enhancement and binarization gives very good results on poor quality images.

Keywords: Fingerprint enhancement, fingerprint binarization, orientation flow.

1 Introduction

One of the most important tasks in an automatic fingerprint recognition system is enhancement of poor quality images [2, 6]. Enhancement leads to

*Supported by a research fellowship from Govt. of India

better accuracy of the system as measured in terms of false rejection and false acceptance rates. Another important problem is binarization of the image into ridges and valleys, which allows for simple and fast binary image processing algorithms [6].

The starting point in most algorithms is that of accurate determination of orientation flow of a fingerprint image. In literature, one of the popular methods to compute the flow is based on gradient analysis, in which one computes second moment gradient matrix, also commonly known as structure tensor, in the neighborhood of each pixel to yield dominant orientation at each pixel [1, 4, 7]. This approach works well if there is indeed a dominant orientation, but which is not the case in many regions of the image, especially if it is of poor quality.

Accurate determination of orientation flow in such problematic regions of fingerprint images is the first major contribution of this paper. Ours is a completely spatial approach which does not use gradients.

In literature, once the orientation flow is determined, authors have proposed several enhancement algorithms. Many use directional filtering and anisotropic diffusion methods [1, 4, 7] or frequency domain methods [6]. Recently, probabilistic techniques have been also proposed [5].

We use directional filtering in our enhancement algorithm. Our filtering is faster as it uses short 1-D Gaussian filter as against 2-D Gaussian or Gabor filters commonly used in the past work. Furthermore, our algorithms for binarization and enhancement fit in the structure of our orientation flow algorithm so that they can be integrated.

Conventionally, the three tasks of orientation flow determination, enhancement and binarization tasks are performed sequentially. However it can be seen that these are mutually dependent on each other. For example, if we had an enhanced image, we could compute more accurate orientation flow, and vice versa.

Motivated by this insight, we propose an iterative scheme and demonstrate that it gives excellent results. One important difference from the past work is that we perform binarization before enhancement. We demonstrate the performance of the proposed projection based iterative algorithm through examples.

2 Second Moment Matrix Based Approach

Let $I(\mathbf{p})$ be gray level intensity of a fingerprint image at location $\mathbf{p} \in \mathbb{R}^2$. Define a neighborhood $\mathcal{N}$ around $\mathbf{p}$. Compute the second moment matrix of gradients of pixel intensity in this neighborhood

$$A = \sum_{\mathbf{q} \in \mathcal{N}} \triangledown I(\mathbf{q}) \triangledown I(\mathbf{q})^T. \tag{1}$$

Let the eigen-values of A be $\lambda_1 \geq \lambda_2 \geq 0$. Let the orientation angle of the dominant eigen-vector $\mathbf{v_1}$ be $\theta = \tan^{-1}(\mathbf{v_1}_y / \mathbf{v_1}_x)$. The angle θ is the orientation direction at pixel p, see [3]. Equivalently,

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{\mathbf{q} \in \mathcal{N}} 2(\triangledown I(\mathbf{q}))_x (\triangledown I(\mathbf{q}))_y}{\sum_{\mathbf{q} \in \mathcal{N}} ((\triangledown I(\mathbf{q}))_x{}^2 - (\triangledown I(\mathbf{q}))_y{}^2)} \right).$$

Often $\mathcal{N}$ is weighted by a Gaussian function so that pixels closer to $\mathbf{p}$ have greater contribution.

We show the problem with this popular approach in Figure 1 (a) in which we show a 25 × 30 subregion of a fingerprint image captured by an optical sensor. At many pixels there is no well defined dominant orientation due to overall poor quality of the image, which then leads to inaccurate orientation flow computation. We will show in next section how we can compute an accurate orientation flow as shown in Figure 1 (b).

3 Projection Based Iterative Algorithm

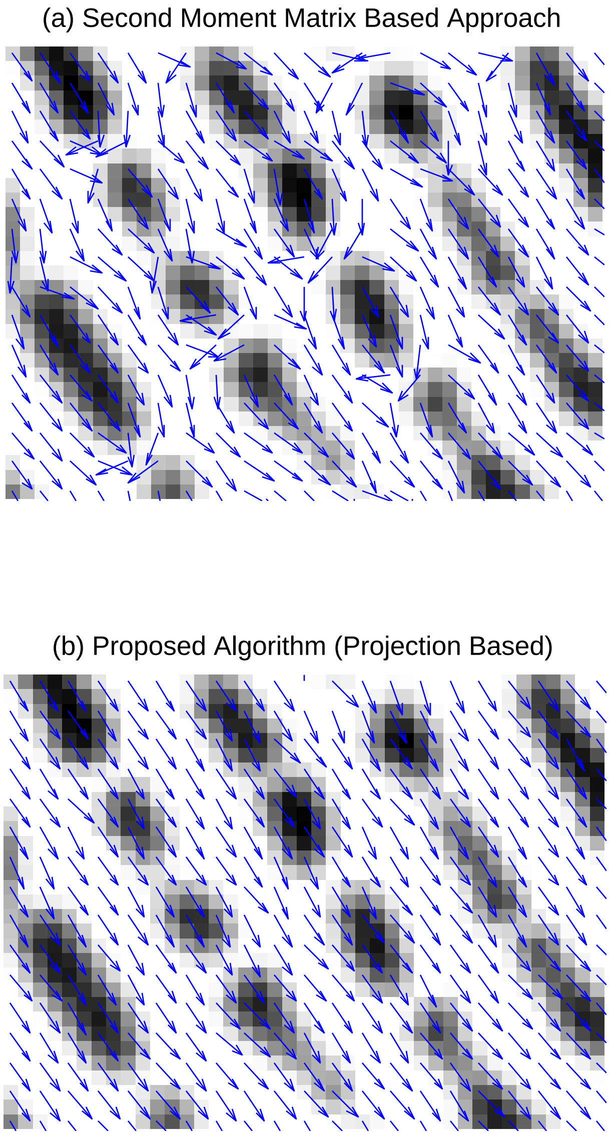
Orientation Flow

Informally, for any point in the image, we take projections in its neighborhood in different directions and find the direction which gives minimum variance in pixel values.

Consider neighborhood $\mathcal{N}$ around $\mathbf{p}$. Let $L(\mathbf{p}, \alpha) \subset \mathcal{N}$ be the line segment passing through $\mathbf{p}$ at an angle α. Let $\mathbf{q} \in L(\mathbf{p}, \alpha)$. Line $L' = L(\mathbf{q}, \alpha + \pi/2) \subset \mathcal{N}$ is the line segment passing through $\mathbf{q}$ which is perpendicular to $L(\mathbf{p}, \alpha)$, see Figure 2 (a).

Denote the standard deviation of pixel values on this perpendicular line as

$$\sigma(\mathbf{q}) = \sigma(L') = \sqrt{\text{Variance}\{I(\mathbf{r}) : \mathbf{r} \in L'\}}.$$



(a) Second Moment Matrix Based Approach

(b) Proposed Algorithm (Projection Based)

Figure 1: Orientation flow determination. (a) Well-known method based on second moment gradient matrix. (b) Our proposed algorithm on the same area of the image.

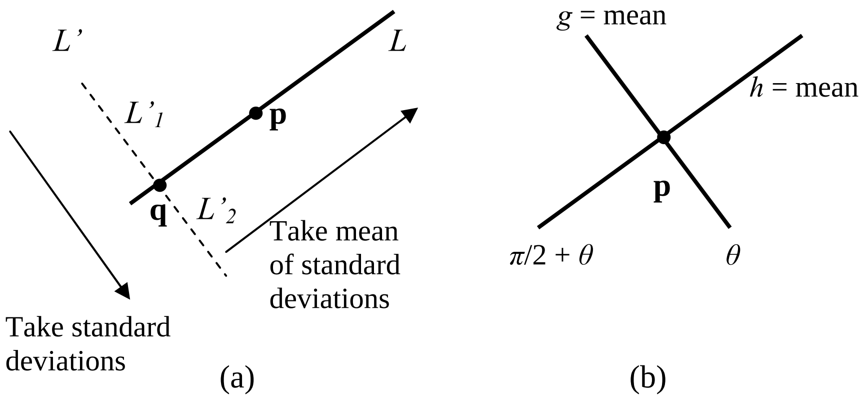


Figure 2: (a) Computation of dominant orientation at pixel $\mathbf{p}$ in the fingerprint image. (b) Binarization of the pixel $\mathbf{p}$.

Take the mean of the standard deviations over all such lines

$$\mu_\alpha = \text{mean}\{\sigma(L') : L' = L(\mathbf{q}, \alpha + \pi/2),\ \mathbf{q} \in L(\mathbf{p}, \alpha)\}.$$

Now we define *dominant orientation* at $\mathbf{p}$ to be the angle which gives minimum average of standard deviations over all angles α, plus $\pi/2$:

$$\theta = \frac{\pi}{2} + \arg\min_{\alpha}\{\mu_\alpha\}. \tag{2}$$

We found that the best results are obtained when we divide the line segment L' into two halves L'_1 and L'_2 which meet at $\mathbf{q}$, as shown in Figure 2 (a), and compute standard deviations for these two line segments, and redefine σ above as

$$\sigma(\mathbf{q}) = \min\{\sigma(L'_1), \sigma(L'_2), \sigma(L')\}.$$

The reason why this gives better result is because even when a point is at the end of a ridge, we need to look at only the half of the neighborhood which is towards the ridge and ignore the other half.

See Figure 1 (b) for results of our proposed algorithm.

We mention few speedup tricks we employed to make the algorithm fast:

1. The orientation flow is not computed at all points $\mathbf{p}$, but only at subsampled set. We computed it at every other pixel in x and y directions yielding speedup factor of 4. At other positions, one can interpolate the angle.

2. The minimum in Equation (2) is obtained by considering α from 0 to π in increments of $\pi/8$. Then we refine this estimate θ by considering smaller increments of $\pi/32$ in the range $[\theta - \pi/16, \theta + \pi/16]$.

3. We rotate the image once for each angle α. Then we pre-compute $I(\mathbf{x})^2$ for each pixel x. This makes computation of standard deviations fast.

Binarization

Informally, to compute the binary value of any pixel one has to check if the greyness values of pixels in the dominant direction is less or more than the values in the orthogonal direction. A related though not same idea is used in [6]. For each point $\mathbf{p}$ with dominant orientation θ, we compute the average greyness values along the line segments passing through it at angles θ and $\theta + \frac{\pi}{2}$, as shown in Figure 2 (b):

$$\begin{aligned} g &= \text{mean}\{I(\mathbf{q}) : \mathbf{q} \in L(\mathbf{p}, \theta)\}, \\ h &= \text{mean}\{I(\mathbf{q}) : \mathbf{q} \in L(\mathbf{p}, \theta + \frac{\pi}{2})\}. \end{aligned}$$

The binary value at p is given as follows

$$B(\mathbf{p}) = \begin{cases} 0, & \text{if } g < h \\ 1, & \text{otherwise} \end{cases}$$

Enhancement

Given the orientation flow and the binary image, we can enhance the image as follows. Informally, for any point we apply a smoothing filter along the dominant direction at those neighboring points which have same binary values as the given point. Consider the sequence of points along line segment $L(\mathbf{p}, \theta)$ which have binary value $B(\mathbf{p})$. Apply a 1-D Gaussian filter on these points to get the enhanced image J as

$$J(\mathbf{p}) = \mathcal{G}(\sigma) * (\mathbf{q} \in L(\mathbf{p}, \theta) : B(\mathbf{p}) = B(\mathbf{q}))$$

where $\mathcal{G}(\sigma)$ is Gaussian filter of standard deviation σ, which we chose to be 3, and which is always normalized to have sum 1, since some $\mathbf{q}$ will not be considered if $B(\mathbf{p}) \neq B(\mathbf{q})$.

Nonlinear Isobrightness Contours

As an advanced option of our algorithm, we generalize the line $L(\mathbf{p}, \theta)$ to a non-linear smooth curve $C(\mathbf{p})$ which is the iso-brightness contour at $\mathbf{p}$. The points just next to $\mathbf{p}$ on this curve are:

$$\begin{aligned}\mathbf{p_1} &= \mathbf{p} + (\cos\theta_{\mathbf{p}}, \sin\theta_{\mathbf{p}}),\\ \mathbf{p_2} &= \mathbf{p} - (\cos\theta_{\mathbf{p}}, \sin\theta_{\mathbf{p}}).\end{aligned}$$

where we denote the dominant orientation at $\mathbf{p}$ by $\theta_{\mathbf{p}}$. The binarization and enhancement algorithm remain same except that we now use $C(\mathbf{p})$. At most pixels, $L(\mathbf{p}, \theta)$ is very good approximation of $C(\mathbf{p})$.

Iterative Algorithm

As we emphasized in the Introduction of this paper, we view orientation flow determination, binarization and enhancement to be three *mutually interdependent* components of overall algorithm, see Figure 3 (b). In Figure 3 (a), we show the conventional scheme. Therefore, we explore this novel approach by applying two iterations of our algorithm i.e. given an image I, once we have computed its orientation flow, binarized it, and enhanced it to give image J, then we repeat the whole process again with the enhanced image J as the input, as illustrated in Figure 3 (c).

4 Results

In Figure 4, we show the results of our algorithm on three different kinds of fingerprint images, captured by optical, capacitive and thermal sweep sensors, see FVC 2002 and 2004 for the datasets [8]. The results on subregions of the images are shown in detail. We used linear approximation to iso-brightness contours for speed and because our experiments indicated that it gives results comparable to the non-linear case. Visual inspection of our results indicates that the algorithm is fairly robust as it works on all three types of sensors. We are able to reproduce the ridges very well, even in the case of capacitive sensor image which is of quite poor quality. One can detect minutiae in binarized images through the operation of morphological skeletonization or some other binary operations.

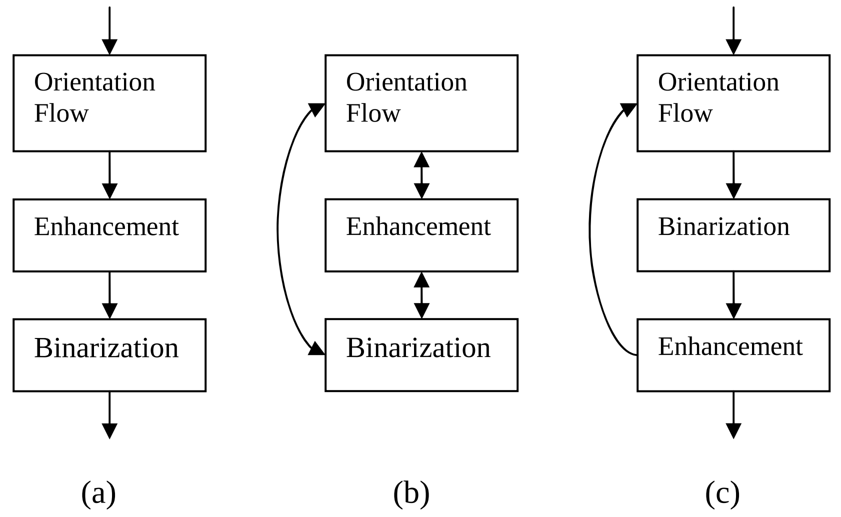


Figure 3: (a) Conventional method to enhance and binarize a fingerprint image. (b) Mutual dependence of subtasks on each other. (c) Our implementation of mutual dependence using an iterative scheme.

5 Conclusions and Future Work

We presented a novel, fast and robust algorithm to accurately compute the orientation flow of a fingerprint image and to binarize and enhance the image. Our algorithm for orientation flow significantly outperforms well-known method based on second moment matrix. We also showed very good results of binarization and enhancement algorithms on images captured by different kinds of sensors.

We believe ideas in this paper can be improved and optimized in future research in several ways. We are currently implementing a complete fingerprint recognition system based on this paper whose results will be described in a forthcoming publication. The mutual interdependence of different components of the algorithm as captured in the iterative algorithm needs to be understood more rigorously on a wide variety of fingerprint images.

References

[1] A. Almansa and T. Lindeberg, “Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators with Automatic Scale Selection,”

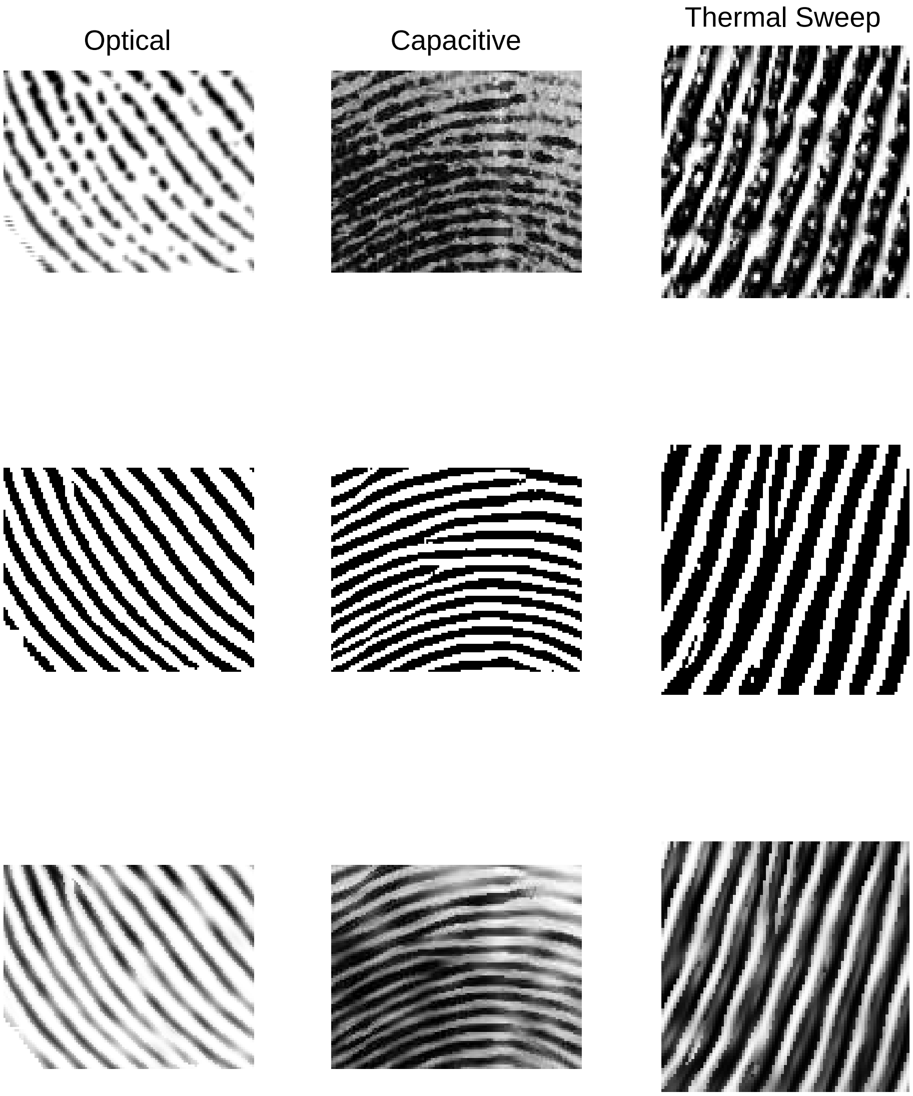


Figure 4: In the left column, we show results on an image captured by an optical sensor, and in the middle column, by a capacitive sensor, and in the right column, by a thermal sweep sensor. We show results in the selected subregions of the images. The top row shows the original images, the middle row shows the binarized images and the bottom row shows the enhanced images.

IEEE Trans. Image Proc., Vol. 9, No. 12, pp. 2027-2042, 2000.

[2] B. Bhanu and X. Tan, *Computational Algorithms for Fingerprint Recognition*, International Series on Biometrics, Vol 1, Springer, 2003.

[3] B. Jähne, *Digital Image Processing,* Springer, 2002.

[4] L. Hong, Y. Wan and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. PAMI*, Vol. 20, No. 8, pp. 777-789, 1998.

[5] K. Lee and S. Prabhakar, "Probabilistic Orientation Field Estimation for Fingerprint Enhancement and Verification," *Biometrics Symposium (BSYM'08)*, pp. 41-46, 2008.

[6] C. I. Watson, *et al.*, "User's Guide to NIST Biometric Image Software (NBIS)," NIST Manuscript, URL: http://fingerprint.nist.gov/NFIS/, 2001.

[7] C. Yu, M. Xie and J. Qi, "An Effective and Robust Fingerprint Enhancement Method," *Proc. of Intnl. Symp. on Computational Intelligence and Design*, 2008.

[8] Fingerprint Verification Competition Datasets, http://bias.csr.unibo.it/fvc2006.