Implementation of An Automatic Fingerprint Identification System

Peihao Huang, Chia-Yung Chang, Chaur-Chin Chen
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 30013
E-mail: cchen@cs.nthu.edu.tw

Outline

- A Flowchart of Fingerprint Identification
  - Type Classification
  - Minutia Points Detection
  - Minutiae Pattern Extraction
  - Pattern Matching
- Rindex28, Lindex101 Databases
- Experimental Results

Flowchart of An AFIS

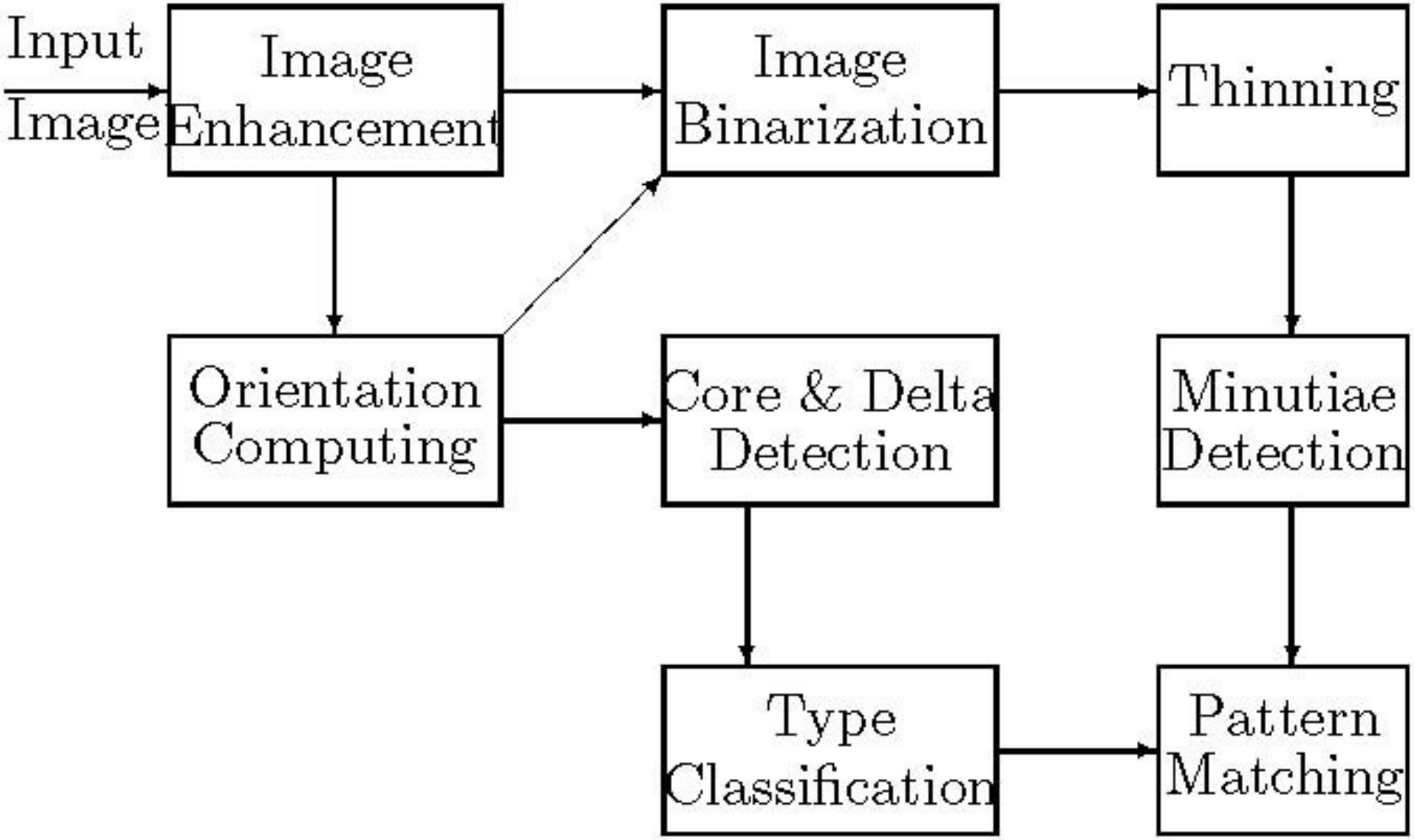


Image Enhancement

- The moisture and scars of a finger as well as the pressure due to a fingerprint sensing could distort the quality of the acquired fingerprint image.
- We adopt an ad hoc strategy to enhance the quality of a fingerprint image.
- Support that A (i, j) is image gray level at pixel (i, j), μ and s^2 are the mean and variance of gray levels of input image, and α=150, γ=95, γ must satisfy γ>s.

  The enhanced image B(i , j) is given as follows.

- $$B(i, j) \leftarrow \alpha + \gamma * ([A(i, j) - \mu] / s)$$

Result of Image Enhancement

Orientation Computation(1)

- First we apply a 5 by 5 median filtering on the image to avoid false gradient vectors generated by noise.
- Then compute the gradient (G_x,G_y) at each pixel by a Sobel operator.

z1	z2	z3
z4	z5	z6
z7	z8	z9

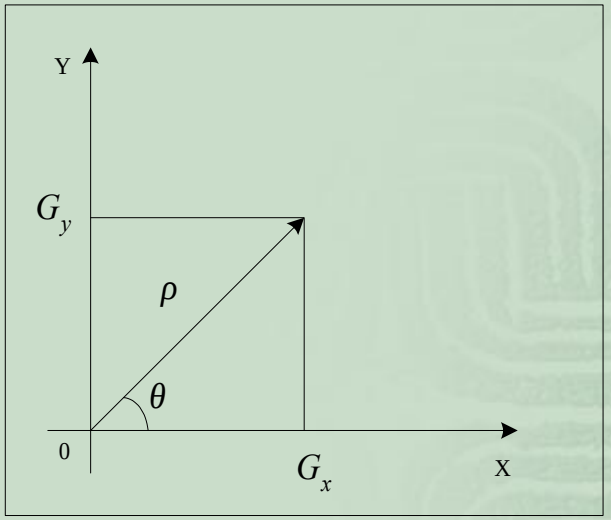
mask

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$
$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Sobel Operation

Orientation Computation(2)

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \rho\cos\theta \\ \rho\sin\theta \end{bmatrix} \qquad \begin{bmatrix} \rho \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{G_x^2 + G_y^2} \\ \tan^{-1}\left(G_y / G_x\right) \end{bmatrix}$$



The relationship between $[G_x, G_y]^T$ and $[\rho,\theta]^T$

Orientation Computation(3)

- Because of opposite gradient vectors might offset each other, we double the angles of the gradient vectors before averaging each block, and let the length of the gradient vectors be squared.
- Let $[\alpha_x, \alpha_y]^T$ be represented by

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} \rho^2 \cos 2\theta \\ \rho^2 \sin 2\theta \end{bmatrix} = \begin{bmatrix} \rho^2 (\cos^2\theta - \sin^2\theta) \\ \rho^2 (2\sin\theta\cos\theta) \end{bmatrix} = \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_xG_y \end{bmatrix}$$

- The average gradient $[\tilde{\alpha}_x, \tilde{\alpha}_y]^T$ in each block R (w×w) is

$$\begin{bmatrix} \tilde{\alpha}_x \\ \tilde{\alpha}_y \end{bmatrix} = \frac{1}{\mathrm{w}^2} \sum_R \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \frac{1}{\mathrm{w}^2} \sum_R \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_xG_y \end{bmatrix}$$

Orientation Computation(4)

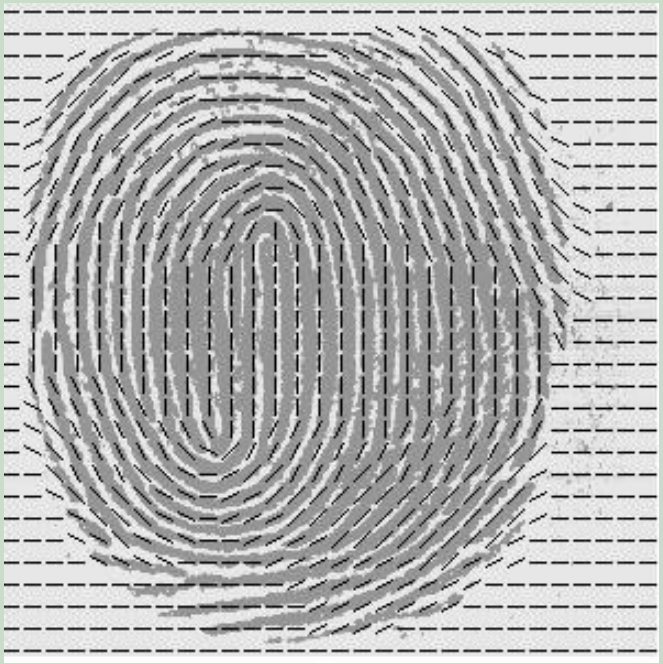
- The block gradient direction ψ is

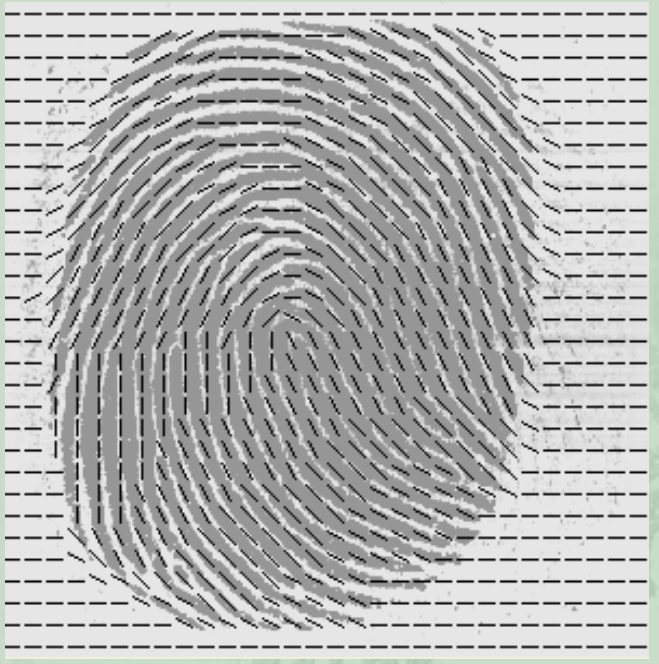
$$\phi = \frac{1}{2}\angle\left(\tilde{\alpha}_x, \tilde{\alpha}_y\right)$$

where $\angle(x, y)$ is defined as:

$$\angle(x,y) = \begin{cases} \tan^{-1}(y/x) & x \geq 0 \\ \tan^{-1}(y/x) + \pi & x < 0 \ \wedge \ y \geq 0 \\ \tan^{-1}(y/x) - \pi & x < 0 \ \wedge \ y < 0 \end{cases}$$

Orientation Computation(5)





Block orientation images

Region of Interest Detection(1)

- To avoid obtaining false singular points or minutiae, we use mean and standard deviation in each block to determine if the block is “good” (not a marginal block) or not.

$$v = \mathrm{w}_0(1-\mu) + \mathrm{w}_1 \times \sigma + \mathrm{w}_2$$

where $\mathrm{w}_0 = 0.5$, $\mathrm{w}_1 = 0.5$ and w_2 is the ratio of distance to the center of the fingerprint image. μ and σ are normalized to be in [0,1].

- If v > 0.8, the block is what we want.

Region of Interest Detection(2)

Enhanced image

Region of interest

Singular Point Detection(1)

- Because of noisy directions, we have to smooth the direction before computing the Poincaré index.
- We regard the direction as a vector, double the angles and use a 3 by 3 averaging filter to smooth the direction.

B_3	B_2	B_1
B_4	B_c	B_0
B_5	B_6	B_7

1	1	1
1	2	1
1	1	1

$$a = \sum_{i=0}^{7} B_{i,x} + 2B_{c,x}$$

$$b = \sum_{i=0}^{7} B_{i,y} + 2B_{c,y}$$

$$B_i = \left(B_{i,x}, B_{i,y}\right), 0 \le i \le 7 \text{ or } i = c$$

- The average direction of the block is $\bar{\theta} = \frac{1}{2}\arctan\left(\frac{b}{a}\right)$

Singular Point Detection(2)

- We compute Poincaré index by summing up the difference in the direction surrounding the block P. For each block P_j, we compute the angle difference from 8 neighboring blocks along counter-clockwise direction.

P_1	P_8	P_7
P_2	P	P_6
P_3	P_4	P_5

$$P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_7 \rightarrow P_8 \rightarrow P_1$$

Core if the sum of difference is 180°

Delta if the sum of difference is -180°

Singular Point Detection(3)

Singular points of Fingerprint Images

4+1 Fingerprint Type Classification

Arch | Left Loop | Right Loop | Whorl | undecided

Type Classification

Type	Arch (tented arch)	Left loop	Right loop	Whorl (twins loop)	Others
# of cores	0 or 1	1	1	2	0 or >2
# of deltas	0 or 1 (middle)	1(right)	1(left)	0~2	0 or >2

Implementation of classification

- All Fingerprint Images are collected by a Veridicom FP110 reader with resolution 500 dpi
- Experiment on Rindex28: 4x28 Right Index Fingerprint Images collected from 28 students
  - No classification error
- Experiment on Lindex101
  - 17 classification errors
    - Due to inappropriately pressing, too complex structure, or poor quality.

Inappropriately Pressing

Whorl

Left Loop

Inappropriately Pressing

Right Loop

Arch

Too Complex Structure

- Left or Whorl
- Left or Arch

Fingerprint Images of Poor Quality

? X ? X

Flowchart of An AFIS

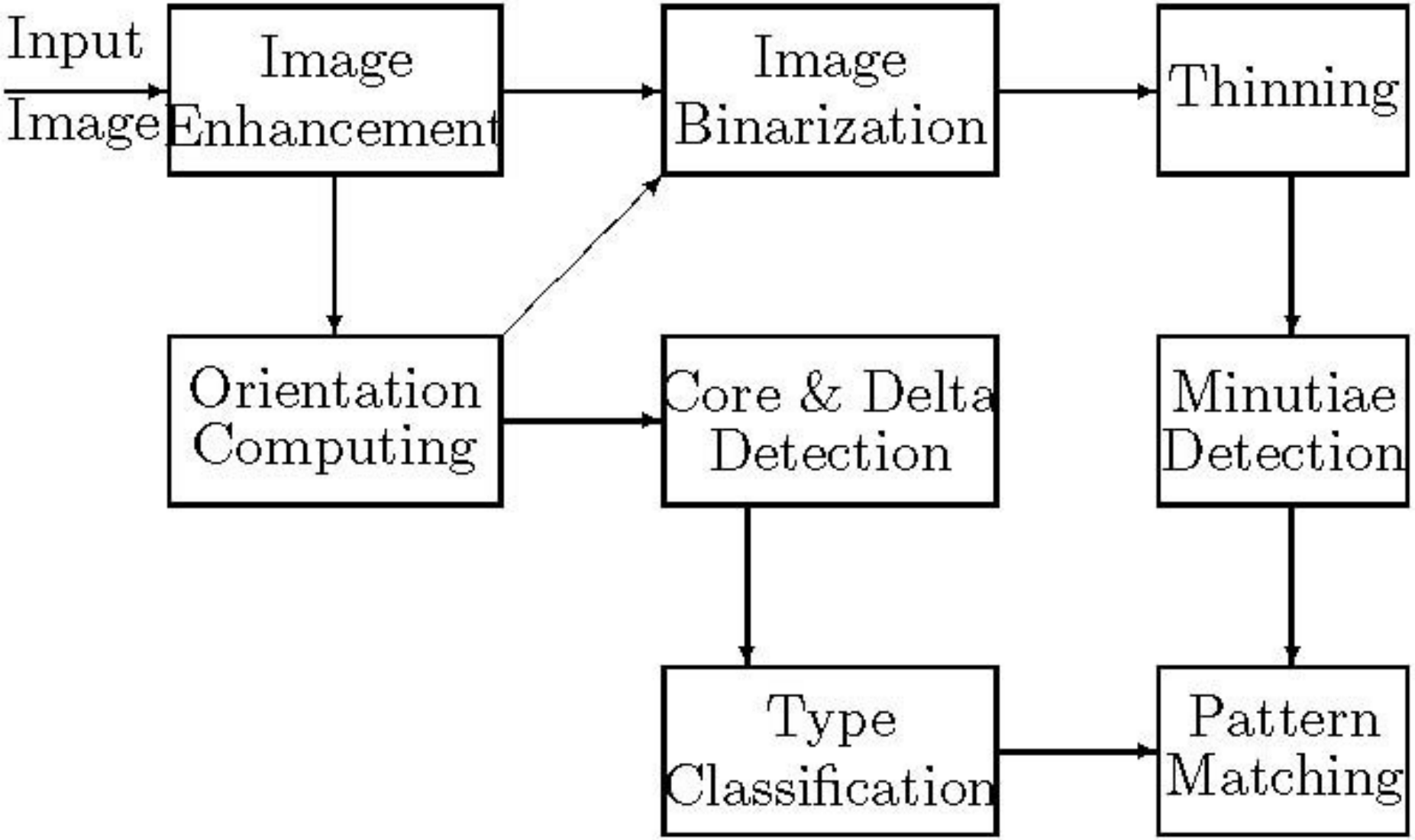


Image Binarization(1)

- We have to distinguish valley and ridge of a fingerprint image before smoothing and thinning. So the gray value of pixels in the enhanced fingerprint image will be binarized to 0 or 255.

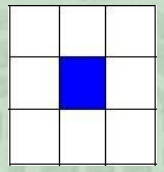
- First we compute the gray value of P_{25} and P_{50} from the enhanced image, where P_k is the kth percentile of enhanced fingerprint image histogram.

- Then we partition an enhanced fingerprint image into w by w blocks and compute the mean of each blocks. We define that M_j is the mean of the j-th block.

Image Binarization(2)

- If the gray value of pixel S_i is less than P_{25}, we assign 0 to S_i . If the gray value of pixel S_i greater than P_{50}, we assign 255 to S_i . Otherwise, the pixel value is defined by the following rule:

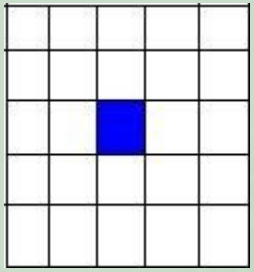
$$S_i = \begin{cases} 255 & \text{if } \frac{1}{8}\sum_{\substack{x=0 \\ x \neq i}}^{8} S_x \geq M_j \\ 0 & \text{otherwise} \end{cases}$$



Smoothing

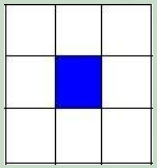
- After binarization, we find that there is still much noise on ridge region. In order to make the result of thinning better, we have to smooth the fingerprint image first. A smooth stage uses neighboring pixels to remove noise.
- First a 5 by 5 filter is used. The pixel p_i is assigned by:

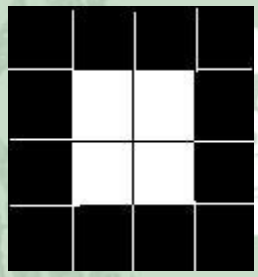
$$p_i = \begin{cases} 255 & \text{if } \Sigma_{5x5} N_w \geqq 18 \\ 0 & \text{if } \Sigma_{5x5} N_b \geqq 18 \\ p_i & \text{otherwise} \end{cases}$$



- Then a 3 by 3 filter is further proceed by:

$$p_i = \begin{cases} 255 & \text{if } \Sigma_{3x3} N_w \geqq 5 \\ 0 & \text{if } \Sigma_{3x3} N_b \geqq 5 \\ p_i & \text{otherwise} \end{cases}$$





(a) Original image

(b) Enhanced image

(c) Binarization image

(d) Smoothed image

Thinning

- The purpose of thinning stage is to gain the skeleton structure of fingerprint image.
- It reduces a binary image consisting of ridges and valleys into a ridge map of unit width.

(d) Smoothed image

(e) Thinned image

Minutiae Detection

- From a thinned image, we can classify each ridge pixel into the following categories according to its 8-connected neighbors.
- A ridge pixel is called :
- an isolated point if it does not have any 8-connected neighbor.
- an ending if it has exactly one 8-connected neighbor.
- an edgepoint if it has two 8-connected.
- a bifurcation if it has three 8-connected.
- a crossing if it has four 8-connected.

Minutiae Extraction

Due to broken ridges, blur effects, and ridge endings near the margins of an image, we have to remove the spurious minutiae as described below.

(1) Two endings are too close (within 8 pixels)

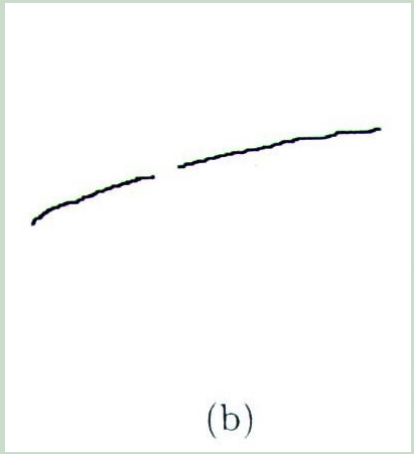
(2) An ending and a bifurcation are too close (< 8 pixels)

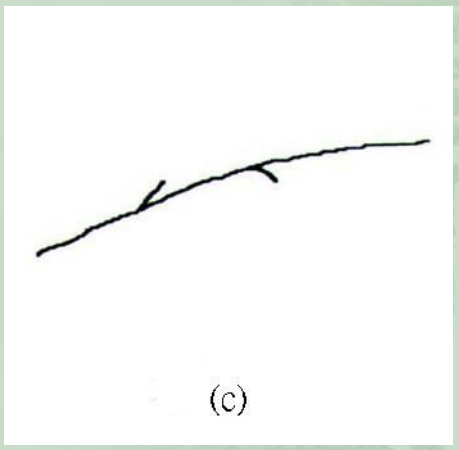
(3) Two bifurcations are too close (< 8 pixels)

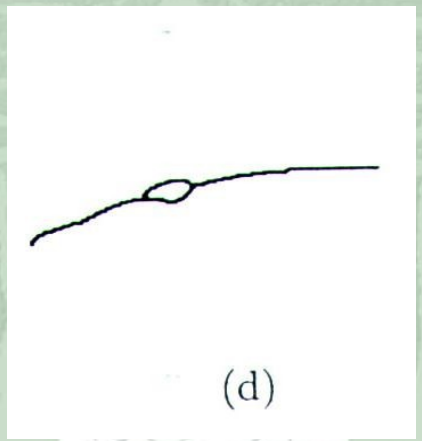
(4) Minutiae are near the margins (< 8 pixels)

Minutiae Extraction

- Spurious minutia pixels include :
  (a) Ending that lie on the margins of the region of interest.
  (b) Two nearest endings with the same ridge orientation.
  (c) ending and bifurcation that are connected and close enough.
  (d) two bifurcations that are too close.







Example of Minutiae Extraction

Example of Minutiae Extraction

Example of Minutiae Extraction

Minutiae Extraction (10)

- Fingerprint Template Data

The information format of fingerprint template data.

Type	#of cores	Core*	# of deltas	Delta*	# of minutiae	Minutiae*
4 bits	2 bits	24 bits	2 bits	24 bits	7 bits	26 bits

The information format of singular points, core or delta.

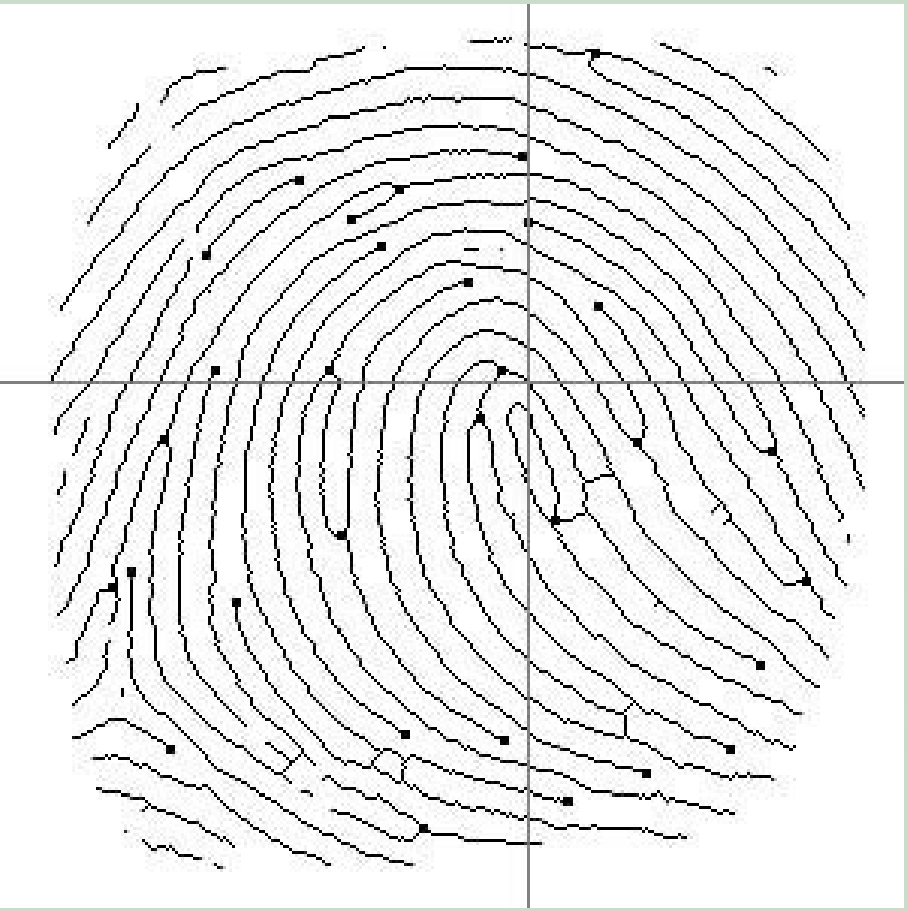
X Coordinate	Y Coordinate	Direction
10 bits	10 bits	4 bits

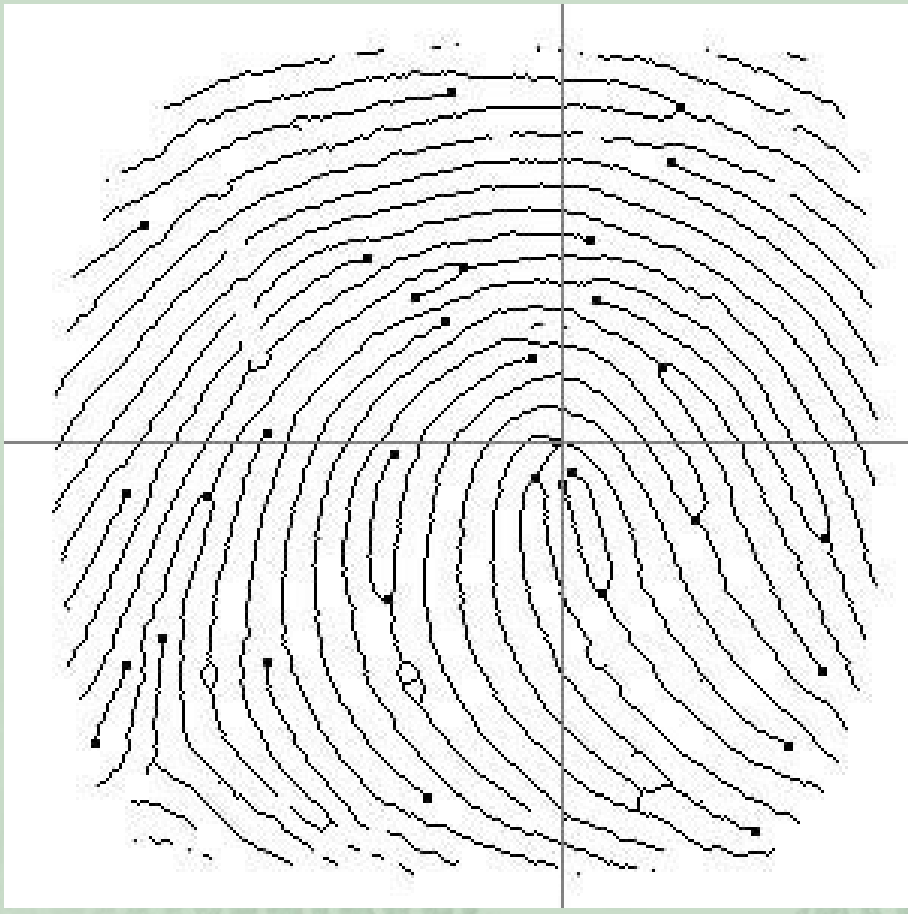
The information format of a minutia.

Kind of Minutiae	X Coordinate	Y Coordinate	Direction
2 bits	10 bits	10 bits	4 bits

Minutiae Matching

Minutiae Pattern Matching





Minutiae Pattern Matching

Fingerprint Matching Score

The matching score of these two fingerprints is calculated by

$$S = 100 \times \frac{1}{M} \sum_{j=1}^{M} \left(1 - \frac{r}{R}\right)_j$$

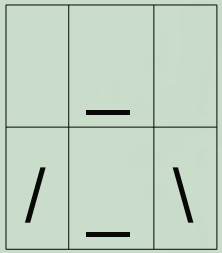
where M is the number of potential type-matching minutiae within a disk of a certain user-specified radius, R (about 8~16 pixels). r measures the distance between a pair of potentially matched minutiae points.

Fingerprint Matching (1)

- Registration Point

  The registration point is regarded as the origin when processing minutiae matching.

  - Left Loop and Right Loop: its core is employed
  - Whorl: the coordinate of the upper-row core is utilized
  - Arch: we apply the mask shown as follows



Fingerprint Matching (2)

- Minutiae Matching

There are four steps involved in our matching process:

(1) Check the type of fingerprint

(2) Overlay by registration point

(3) Rotate and relocate

(4) Compute the matching score

(5) Comparison (the larger match score, the better match)

Fingerprint Matching (3)

The matching score of these two fingerprints is calculated by

$$S = 100 \times \frac{1}{M} \sum_{j=1}^{M} \left(1 - \frac{r}{R}\right)_j$$

where M is the number of potential type-matching minutiae within a disk of a certain user-specified radius, R (about 8~16 pixels). r measures the distance between a pair of potentially matched minutia points.

Fingerprint Database (1)

- Rindex28

  Rindex28, is obtained from PRIP Lab at NTHU. It contains 112 images of size 300 by 300 contributed by 28 different individuals. Each contributed 4 times with the same right index finger scanned by a Veridicom FPS110 live scanner with 500 dpi

Fingerprint Database (2)

- Lindex101

  Lindex101, is obtained from PRIP Lab at NTHU. It contains 404 images of size 300 by 300 contributed by 101 different individuals. Each contributed 4 times with the same left index finger scanned by a Veridicom FPS110 live scanner with 500 dpi

Fingerprint Database (3)

- FVC2000

	Sensor Type	Image Size	Resolution
DB1	Low-cost Optical Sensor	300x300	500 dpi
DB2	Low-cost Capacitive Sensor	256x364	500 dpi
DB3	Optical Sensor	448x478	500 dpi
DB4	Synthetic Generator	240x320	about500 dpi

Fingerprint Database (4)

Examples of fingerprint images from each database of FVC2000

Experimental Results

	Rindex28	Lindex101	DB1	DB2	DB3	DB4
Recognition rate	99.11% 111/112	82.67% 334/404	92.50% 74/80	90.00% 72/80	87.50% 70/80	92.50% 74/80
Enrolling time for each fingerprint image	0.25 sec	0.25 sec	0.25 sec	0.25 sec	0.45 sec	0.17 sec
Matching time	0.359 sec	3.14 sec	0.25 sec	0.218 sec	0.234 sec	0.156 sec

The experimental results of 6 databases

Mismatched Fingerprint Images

Conclusion

€ We reveal three problems, which affect the result of an AFIS which merit further studies.

(1) Noise produces the poor binarization results

(2) Broken ridges result in the error orientation, which causes the misclassification of a fingerprint type

(3) The shifted fingerprint image is difficult to match the minutiae pattern well, for example, the type misclassification due to the missing cores or deltas