Combinatorial Classification of Pixels for Ridge Extraction in a Gray-scale Fingerprint Image*

A. Bishnu, P. Bhowmick, J. Dey, B. B. Bhattacharya†, M. K. Kundu, C. A. Murthy
{bishnu_t, partha_t, bhargab, malay, murthy}@isical.ac.in
Indian Statistical Institute, 203 B. T. Road, Calcutta - 700108
and
T. Acharya
tinku_acharya@ieee.org
Elution Technology Inc., Phoenix, Arizona, USA

Abstract

Automatic Fingerprint Identification Systems (AFIS) are usually based on minutiae matching. Minutiae are the terminations and bifurcations of the ridge lines in a fingerprint image. Detection of ridge lines from a noisy gray-scale fingerprint image is a challenging task. In this work, a novel combinatorial approach is proposed for classifying each pixel into one of the three classes (crest, valley, and slope) based on its gray-scale topographical relationship with its neighbors. A two-pass algorithm is developed for the pixel classification scheme. The ridge lines are then detected as formed by the thinned version of the crest pixels. The algorithm is robust, performs very well in the presence of noise, and has minimal dependence on thresholding. It has been tested on several fingerprint images in the NIST Special Database 14 and NIST Special Database 4, and is observed to produce good results both in terms of quality of solutions and CPU time.

1. Introduction

In a fingerprint image, the spatial distribution of gray-level intensity values can be understood from the ridge and valley classification of pixels. A ridge (or valley) occurs when there is a connected sequence of pixels having gray-tone intensity values that are significantly higher (or lower) in the sequence than those pixels in the neighborhood of the sequence [5]. In [5], ridges and valleys in a digital image were found by looking for zero-crossings of the first directional derivative in a suitable direction. Ridge like structures in a digital image can also be extracted by convolving the image with different derivatives of Gaussians. In [10], two ridgeness measuring differential operators were studied with respect to their usability in CT/MRI matching of human brain scans. López et al. in [9] discusses the use of some discrete multilocal measures for ridge finding.

Automatic fingerprint identification systems (AFIS) are a class of biometric techniques widely used for personal identification. They are usually based on minutiae matching [3, 6, 7, 8]. Minutiae, or Galton's characteristics [4] are local discontinuities in terms of terminations and bifurcations of the ridge lines that constitute a fingerprint pattern. These two types of minutiae are considered by Federal Bureau of Investigation for identification purposes [18]. AFIS based on minutiae matching involve different stages:

1. fingerprint image acquisition,
2. preprocessing of the fingerprint image,
3. feature extraction (e.g. minutiae) from the image,
4. matching of fingerprint images for identification.

In [13], Mehtre described the steps for handling noise in a fingerprint image, enhancement and restoration of the image, and a parallel thinning procedure. A detailed discussion on all the aspects of personal identification using fingerprints as an important biometric technique can be found in Jain et al. [7]. In their method, a segmentation algorithm based on the local certainty level of the estimated orientation field of the fingerprint image is used to locate the region of interest. Ridges in these zones are then detected by convolving the original image with two masks for increasing the local maximum gray value along a direction normal to the local ridge direction. In O'Gorman and Nickerson's [14] work, a $k \times k$ spatial filter mask, designed based on user inputs, is used with an appropriate orientation for labeling the pixels as foreground (crest) or background. Thinning is done on the binary image obtained before minutiae extraction. In [12], Mehtre and Chatterjee described a method of segmenting a fingerprint image into ridge zones and background based on some statistics of the local orientations of the ridges of the original image. A gray-scale variance method is used in the image blocks having uniform gray-

*This work is funded by a grant from Intel Corp., USA (PO #CAC042717000)

†Author for correspondence.

level where the directional method of segmentation fails. A one-pixel thick skeletonized binary image obtained from ridge lines are used for minutiae extraction. Thinning is a useful preprocessing step to transform a digital image to a 1-pixel wide skeletonized image so that the significant features of the original image are retained and highlighted [16]. A recent and comprehensive work for extracting minutiae from such a binary image is due to Farina et al. [3]. Deriving a one-pixel thick binary image from the original gray level image is a difficult task as noise and different levels of contrast in the image may produce false minutiae or hide real minutiae. Lately, Maio and Maltoni [11] have developed a gray level ridge line tracing algorithm for minutiae extraction directly from the gray scale domain. The algorithm treats a fingerprint image as a gray level zone of ridges and background, whereas, such an image actually consists of three regions - ridges, valleys, and background. Based on the three regions of an image, a statistical analysis of the gray level histogram is used to extract the global information about the range of ridges, valleys and background [2]. A robust ridge detection procedure for minutiae extraction should not miss any ridge. False ridges leading to spurs and bridges [3] may be taken care of by a preprocessing stage prior to minutiae extraction.

In this work, thinning of a gray-level fingerprint image is viewed as a twofold process: a new combinatorial pixel classification scheme for ridge extraction followed by a binary thinning of the ridges. Ridges or crests (henceforth the term ridge or crest will be used interchangeably) are extracted from a fingerprint image by classifying each pixel combinatorially. A two-pass algorithm is developed to classify a pixel into three classes, namely crest (CR), valley (VA), or slope (SL). The proposed crest finding algorithm is based on the use of a Look-Up-Table(LUT). The crest pixels necessarily trace out the ridge lines in the image. Finally, any standard binary thinning algorithm can be applied on the crest pixels as object and the rest as background to extract the one-pixel thick ridge lines.

2. Theme

2.1 Directions and gradients

Let I be an $N \times M$ gray-scale image with g gray levels. Let $L(i,j)$ denote the intensity level of a pixel P. Denote the boundary pixels $\mathcal{B}$ of P as $\{p_{i-1,j},\ p_{i+1,j},\ p_{i-1,j-1},\ p_{i+1,j+1},\ p_{i,j-1},\ p_{i,j+1},\ p_{i+1,j-1},\ p_{i-1,j+1}\}$ respectively with corresponding directions {N, S, NW, SE, W, E, SW, NE} as shown in Figure 1. We fix the direction of a straight line walk from each point in $\mathcal{B}$ to another point in $\mathcal{B}$ through P. It is obvious that 4 such directions are possible (see Figure 1). The directions $\mathcal{D}$, are $(k,l) \in \{(N,S),(NW,SE),(W,E),(SW,NE)\}$, where (k,l) is a fixed direction of walk from k to l. We define two types of elementary walks, w_1 and w_2. A movement from any point in $\mathcal{B}$ to P is defined as w_1; w_2 is defined as a movement from P to any point in $\mathcal{B}$. A walk along any of the directions of $\mathcal{D}$ consists of a walk w_1 followed by a walk w_2 in the same direction (e.g., a walk along (N,S) consists of a walk w_1 from $p_{i-1,j}$ to P followed by a walk w_2 from P to $p_{i+1,j}$). Note that the sign of the gradient changes for walks w_1 and w_2 if there is a crest (valley) (See Figure 1) in a particular direction. The gradient along w_1 is measured as the difference between $L(i,j)$ and the intensity level at any point on $\mathcal{B}$, where w_1 starts. Similarly, the gradient along w_2 is the difference between the intensity level at any point on $\mathcal{B}$ and $L(i,j)$ e.g., along $(k,l) = (N,S)$, the gradient for w_1 is measured as $L'_{w_1} = L(i,j) - L(i-1,j)$ and for w_2, it is measured as $L'_{w_2} = L(i+1,j) - L(i,j)$. We define first difference pairs along any direction $(k,l) \in \mathcal{D}$ as $\Delta_{(k,l)} = \{L'_{w_1,(k,l)}, L'_{w_2,(k,l)}\}$; and $sign(\Delta_{(k,l)}) = \{sign(L'_{w_1,(k,l)}), sign(L'_{w_2,(k,l)})\}$, where,

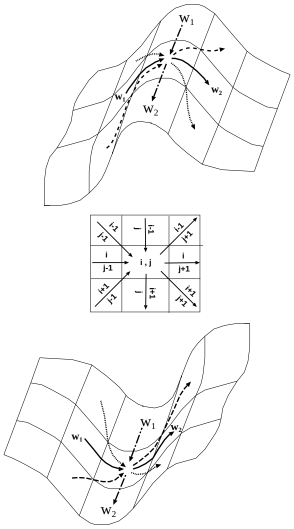


Figure 1: Intensity profiles of crest and valley.

$$
\begin{aligned}
sign(L'_{w_x,(k,l)}) &= +, \text{if } L'_{w_x,(k,l)} > 0,\\
&= -, \text{if } L'_{w_x,(k,l)} < 0,\\
&= 0, \text{if } L'_{w_x,(k,l)} = 0, \text{for } x = 1,2.
\end{aligned}
$$

2.2 Combinatorial possibilities

For a point to be a crest (CR) along any particular direction $\mathcal{D}_{(k,l)}$, there can be three cases:

(i) w_1 reaches a crest and w_2 falls off signifying a change in the sign of gradient, i.e., $sign(\Delta_{(k,l})=\{+,-\}$;

(ii) w_1 reaches a crest and w_2 is on the crest with no gradient change, i.e., $sign(\Delta_{(k,l})=\{+,0\}$;

(iii) w_1 is on the crest with no gradient change, and w_2 falls off the crest, i.e., $sign(\Delta_{(k,l})=\{0,-\}$.

Similarly for a valley (VA), the three cases are:

(i) w_1 reaches a valley and w_2 rises signifying a change in the sign of gradient, i.e., $sign(\Delta_{(k,l)} = \{-,+\}$;
(ii) w_1 reaches a valley and w_2 is on the valley with no gradient change, i.e., $sign(\Delta_{(k,l)} = \{-,0\}$;
(iii) w_1 is on the valley with no gradient change, and w_2 rises from the valley, i.e., $sign(\Delta_{(k,l)} = \{0,+\}$.

A pixel P that is on the slope of the intensity landscape is called a slope (SL) point. Obviously, there would be no change in the sign of gradients:
(i) $sign(\Delta_{(k,l)} = \{+,+\}$, (ii) $sign(\Delta_{(k,l)} = \{-,-\}$.

See Figures 1 and 2 for the possibilities along the directions. Note that, considering pairs from the set $\{+,-,0\}$, we have $3^2 = 9$ cases out of which 8 (3 cases of CR, 3 cases of VA, and 2 cases of SL) have been taken care of, and for the last case, i.e., where $sign(\Delta_{(k,l)} = \{0,0\}$, we label P as undecidable (UN).

Along any direction $(k,l) \in \mathcal{D}$, we can label P from any one of the elements of set $C = \{CR, VA, SL, UN\}$, i.e. $C_{k,l}(P) = CR/VA/SL/UN$. If P is found to be a crest along majority of the directions $(k,l) \in \mathcal{D}$, then there is a high probability of P to be a crest. Thus, we define the label of $C(P)$ as: $C(P) = f_{(k,l)\in\mathcal{D}}\{C_{(k,l)}(P)\}$, where f is a function as follows: $f : C_{(N,S)}(P) \times C_{(NW,SE)}(P) \times C_{(W,E)}(P) \times C_{(SW,NE)}(P) \rightarrow C(P)$. As an example, we can take a majority vote among the different directions to finally assign P to any element from the set C, i.e., $C(P) = max_{(k,l)\in\mathcal{D}}\{C_{(k,l)}(P)\}$. The upper bound on the number of combinatorial possibilities of the elements of C along directions $(k,l) \in \mathcal{D}$ is $4^4 = 256$, with 4 possibilities along each of the 4 directions. Since, the definition of $C(P)$ does not take into account the effect of directionality, several combinations become identical (e.g., 2 CRs along *any* 2 directions are the same). To tighten the count, we define $x_i, (i \in C)$, as the number of directions having the label 'i'. Clearly, x_i can take integral values in $[0,4]$. Further, the total number of directions is bounded by 4. Thus, $\sum_{i\in C} x_i = x_{CR} + x_{VA} + x_{SL} + x_{UN} = 4$. Finding the number of possible integral solutions of the above equation is equivalent to finding the coefficient of x^4 in the *generating function* [15] $(x^0 + x^1 + x^2 + x^3 + x^4)^4$

= coefficient of x^4 in $(1-x)^{-4}(1-x^5)^4 = 35.$

Fact 1 The number of combinatorial possibilities, thus determined, depends on the cardinality of the set $\mathcal{D}$ ($(k,l) \in \mathcal{D}$) of directions and not on the set $\mathcal{B}$ of boundary pixels. Generalizing, if n directions are chosen and along each direction the number of possibilities is 4, then the number of combinatorial possibilities is the coefficient of x^n in $(1-x)^{-4}(1-x^{n+1})^4 = (n+1)(n+2)(n+3)/6$. Thus, the size of the Look-Up-Table (LUT) is $(n+1)(n+2)(n+3)/6$ $= O(n^3)$. In our case, the LUT will therefore, have 35 rows (see Table 1). $\square$

3. Classification of a pixel

3.1 Classification along a direction

The relative gray-scale topographical configuration of L in its locality can be viewed from four possible directions as shown in Figure 1. To calculate the first difference pairs along the walks w_1 and w_2 defined in Section 2, we take directional averages along the directions $(k,l) \in \mathcal{D}$ for calculating L'_{w_1} and L'_{w_2}. The calculation of L'_{w_1} and L'_{w_2} at P using directional averages for a 5×5 neighborhood along a direction (NW, SE) is as follows:

$L'_{w_1,(NW,SE)} =$
$L(i,j) - (L(i-1,j-1) + L(i-2,j-2))/2,$
$L'_{w_2,(NW,SE)} =$
$(L(i+1,j+1) + L(i+2,j+2))/2 - L(i,j).$

Consider a neighborhood of P as defined above and using directional averages, the first difference pairs are:

$\Delta_{(N,S)} = (a,b)$:
$a = L(i,j) - (L(i-1,j) + L(i-2,j))/2,$
$b = (L(i+1,j) + L(i+2,j))/2 - L(i,j),$
$\Delta_{(NW,SE)} = (c,d)$:
$c = L(i,j) - (L(i-1,j-1) + L(i-2,j-2))/2,$
$d = (L(i+1,j+1) + L(i+2,j+2))/2 - L(i,j),$
$\Delta_{(W,E)} = (e,f)$:
$e = L(i,j) - (L(i,j-1) + L(i,j-2))/2,$
$f = (L(i,j+1) + L(i,j+2))/2 - L(i,j),$
$\Delta_{(SW,NE)} = (g,h)$:
$g = L(i,j) - (L(i+1,j-1) + L(i+2,j-2))/2,$
$h = (L(i-1,j+1) + L(i-2,j+2))/2 - L(i,j).$

Directional averages are considered for noise immunity which is a desirable property [9] of a ridge finding method. Each of the eight parameters $a, b, c, \ldots, h$, can either be positive, or negative, or zero. Now, based on the definition of walks w_1 and w_2 in the previous section, we consider the pairwise property of the eight parameters (i.e., a and b, c and d, e and f, g and h) and, therefore, each pair of parameters along the direction of the walks can have 9 possibilities as shown in Figure 2.

The fourth column ('Gray Levels'), in Figure 2 exhibits various pictorial representations of the three points P, P', P'', where P' and P'' are the adjacent pixels of P in the concerned pair of parameters. The values of these four pairs indicate the topographical configuration of P in its locality. As shown in the rightmost column of Figure 2, the $sign$ (as defined in Section 2.1) of each of the four pairs of parameters ((a,b), (c,d), (e,f), (g,h)) is used to assign a class out of the four classes CR, VA, SL and UN, to the respective pixel P. Thus, each pixel P is assigned to any one of the four preliminary classes along a single direction. After the preliminary classification pass, P can be either strongly classified or weakly classified as discussed next.

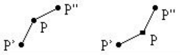
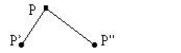
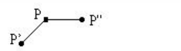
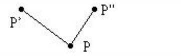
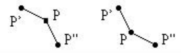
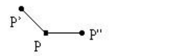
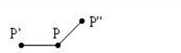
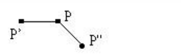
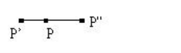
CASE	a	b	GRAY LEVELS	CLASS OF P
1	+	+		SL
2	+	-		CR
3	+	0		CR
4	-	+		VA
5	-	-		SL
6	-	0		VA
7	0	+		VA
8	0	-		CR
9	0	0		UN

Figure 2: Relative position of P w.r.t. P' and P''.

3.2 Preliminary classification of a pixel

For preliminary classification of a pixel P, we implement the function $C(P) = f_{(k,l)\in\mathcal{D}}\{C_{(k,l)}(P)\}$ as a Look-Up-Table with a bias towards crest as shown in Table 1. This bias stems from the need of fingerprint analysis as the minutiae are defined as the discontinuities on the ridge lines. The cases that cannot be topographically classified are processed further. We define four intermediate classes CV, CS, VS and XX; they include the unresolvable pixels requiring further processing. Thus, a pixel P is classified to one of the classes among CR, VA, SL, CV (can be crest or valley), CS (crest or slope), VS (valley or slope), XX (crest, valley or slope) depending on their topographical configuration. The exhaustive 35 cases (as deduced in Section 2.2) are shown in Table 1. It may be observed that even if along any direction at least one CR is present, it is labeled as either CV, CS or XX, thus keeping the option of being classified as CR at the next stage of classification.

3.3 Final classification

In Table 1, the criteria of strong classification is not satisfied by the ambiguous classes CV, VS, CS and XX. These pixels are finally classified in the second pass by inspecting the presence of other pixels that are classified unambiguously in its neighborhood. This method is similar to the multilocal approach suggested in [9]. For a pixel P belonging to CV, CS, VS or XX, we define a neighborhood $\mathcal{R}(P)$ whose size is determined by certain criteria as discussed in Section 3.3.1. Let in $\mathcal{R}(P)$, the average gray value of the pixels belonging to CR be $Ave(CR)$. Similarly, let $Ave(VA)$ and $Ave(SL)$ be the average gray value of pixels classified as valley and slope respectively. For any pixel $P \in CV$, we classify it to either crest or valley based on the closeness of the gray value $L(i,j)$ of P to $Ave(CR)$ or $Ave(VA)$, i.e., if $|L(i,j) - Ave(CR)| \leq |L(i,j) - Ave(VA)|$, then assign P to CR, else assign it to VA. Similarly, assign CS to CR or SL; and VS to VA or SL. The pixel $P \in XX$ is assigned to either CR, VA or SL for which its gray-value difference is minimum.

Table 1: Classification of a pixel

No. of cases	No. of classes in preliminary set				Preliminary Class
	CR	VA	SL	UN	
1	0	0	0	4	XX
2	0	0	1	3	XX
3	0	0	2	2	XX
4	0	0	3	1	SL
5	0	0	4	0	SL
6	0	1	0	3	XX
7	0	1	1	2	XX
8	0	1	2	1	XX
9	0	1	3	0	VS
10	0	2	0	2	VA
11	0	2	1	1	VA
12	0	2	2	0	VS
13	0	3	0	1	VA
14	0	3	1	0	VA
15	0	4	0	0	VA
16	1	0	0	3	XX
17	1	0	1	2	XX
18	1	0	2	1	CS
19	1	0	3	0	CS
20	1	1	0	2	CV
21	1	1	1	1	CV
22	1	1	2	0	CV
23	1	2	0	1	CV
24	1	2	1	0	CV
25	1	3	0	0	VA
26	2	0	0	2	CR
27	2	0	1	1	CR
28	2	0	2	0	CR
29	2	1	0	1	CR
30	2	1	1	0	CR
31	2	2	0	0	CV
32	3	0	0	1	CR
33	3	0	1	0	CR
34	3	1	0	0	CR
35	4	0	0	0	CR

3.3.1 Determination of $\mathcal{R}(P)$

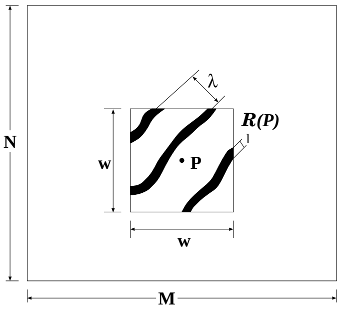


Figure 3: Determination of $\mathcal{R}(P)$.

Let $\mathcal{R}(P)$ be a square region of size $w \times w$ centered at P. Let the number of ridges be approximately k in a typical fingerprint. The size of the fingerprint image is $N \times M$. Let l be the width of the ridges in terms of pixels. Thus, the image size can be estimated in terms of k, l and the inter-ridge distance λ. Therefore, the approximate inter-ridge distance λ can be estimated roughly as (see Figure 3)

$$\begin{aligned}(\lambda + l)k &= \sqrt{N^2 + M^2} \\ \text{or } \lambda &= \frac{\sqrt{N^2 + M^2}}{k} - l\end{aligned}$$

The size of $\mathcal{R}(P)$ should be such that it includes at least some of the crest lines. To include r crest lines, w should therefore, be equal to $(r\lambda)/\sqrt{2}$.

4 Evaluation and results

After the final classification, the image reduces to 2/3-pixel thick crest lines, valley lines, and the rest consisting of slope regions, (see Figure 4(b)). The valley and slope pixels are treated as background, and the crest pixels are treated as foreground object. To obtain one-pixel thick crest lines, the resulting binary image can now be thinned by any standard thinning technique reported in literature [16]. Results shown in Figures 4-6 were obtained by running our algorithm on some image samples from the NIST Special Database 14 [1] and NIST Special Database 4 [17]. The samples are of size 480×512 with 500 dpi resolution and 256 gray levels. Figure 4 shows the original image, classified ternary image and the thinned ridge lines for one sample. Figures 5(a) and 6(a) show the original image, and Figures 5(b) and 6(b) show the thinned ridge line superimposed on the original image for two samples from the NIST 14 and NIST 4 databases. The proposed algorithm was implemented in C on a Sun_Ultra 5_10, Sparc, 233 MHz, the OS is SunOS Release 5.7 Generic. The total CPU time for the entire classification was found to be on an average 2.5 secs.

5 Conclusions and discussions

In this work, we have enumerated the exhaustive combinatorial possibilities that a particular pixel can have in a digital image landscape in terms of the first difference pairs. Based on the possibilities, the pixels are classified into three different classes using a two-pass algorithm. The combinatorial possibilities stored as a LUT provide us an efficient tool for VLSI implementation of the first pass of the classification scheme on-chip. The construction of the LUT is based on some empirical observations, and relevant experiments have been performed on the NIST14 and NIST4 databases. The proposed method provides a very convenient technique for extracting ridge lines from a fingerprint image. A better design of the LUT including magnitudes in addition to signs needs further investigation. The generating function has to be suitably tailored for calculating the combinatorial possibilities accordingly.

References

[1] 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*, NISTIR 5647, National Institute of Standards and Technology, August 1995.

[2] Chang, J. -H., Fan, K. -C., "Fingerprint Ridge Allocation in Direct Gray-Scale Domain", *Pattern Recognition*, vol. 34, no. 10, pp. 1907-1925, Oct., 2001.

[3] Farina, A., Zs. M. Kovács-Vajna, Zs., M. and Leone, A., "Fingerprint Minutiae Extraction from Skeletonized Binary Images", *Pattern Recognition*, vol. 32, pp. 877-889, 1999.

[4] Galton, F., "Fingerprints", *London: Macmillan*, 1892.

[5] Haralick, R., "Ridges and Valleys on Digital Images", *Computer Vision Graphics Image Processing*, vol. 22, pp. 28-38, 1983.

[6] Hollingum, J., "Automated Fingerprint Analysis Offers Fast Verification", *Sensor Review*, vol. 12, no. 13, pp. 12-15, 1992.

[7] Jain, A. K., Hong, L., Pankanti, S. and Bolle, R., "An Identity-Authentication System Using Fingerprints" , *Proc. of IEEE*, vol. 85, no. 9, pp. 1365-1388, Sep., 1997.

[8] Kovács-Vajna, Z. M., "A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping", *IEEE Trans. PAMI*, vol. 22, no. 11, pp. 1266-1276, Nov., 2000.

[9] López, A. M., Lumbreras, F., Serrat, J. and Villanueva, J. J., "Evaluation of Methods for Ridge and Valley Detection", *IEEE Trans. PAMI*, vol. 21, no. 4, pp 327-335, April, 1999.

[10] Maintz, J. B. A., Elsen, van den P. A. and Viergever, M. A., "Evaluation of Ridge Seeking Operators for Multimodality Medical Image Matching", *IEEE Trans. PAMI*, vol. 18, no. 4, pp. 353-365, April, 1996.

[11] Maio, D. and Maltoni, D., "Direct Gray-Scale Minutiae Detection In Fingerprints", *IEEE Trans. PAMI*, vol. 19, no. 1, pp. 27-39, 1997.

[12] Mehtre, B. M. and Chatterjee, B., "Segmentation of Fingerprint Images - A Composite Method", *Pattern Recognition*, vol. 22, pp. 381-385, 1989.

[13] Mehtre, B. M., "Fingerprint Image Analysis for Automatic Identification", *Machine Vision and Applications*, vol. 6, no. 2, pp. 124-139, 1993.

[14] O'Gorman. L. and Nickerson, J. V., "An Approach to Fingerprint Filter Design", *Pattern Recognition*, vol. 22, pp. 29-38, 1989.

[15] Roberts, F. S., *Applied Combinatorics*, Prentice Hall Inc., Englewood Cliffs, NJ, 1984.

[16] Rosenfeld, A. and Kak, A. C., *Digital Image Processing*, vol. 2, Academic Press Inc., Orlando, Florida, 1982.

[17] Watson, C. I., Wilson, C. L., *Fingerprint Database*, National Institute of Standards and Technology, Special Database 4, FPDB, April, 1992.

[18] Wegstein, J. H., "An Automated Fingerprint Identification System", *US Government Publication*, Washington, 1982.

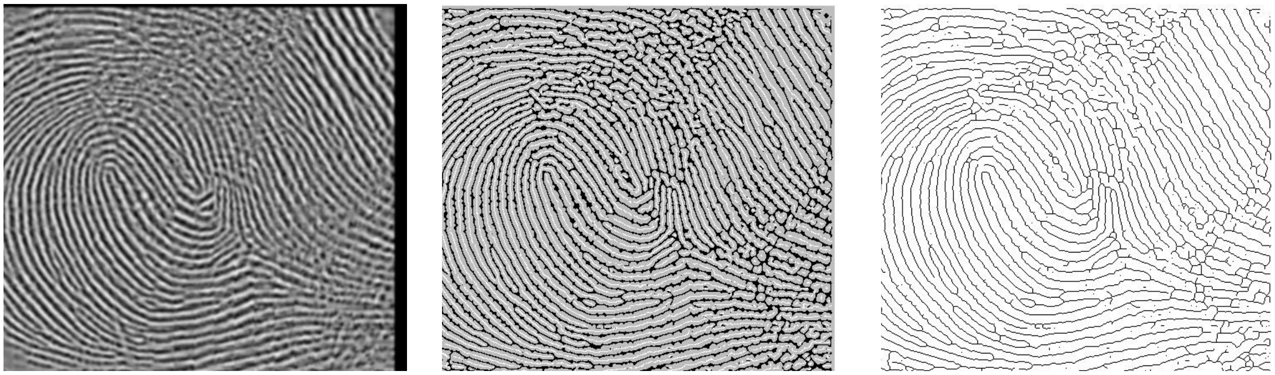

(a) Original image (b) Ternary image (c) Binary image

Figure 4: A fingerprint image sample from NIST 14 sdb showing the stages of classification.

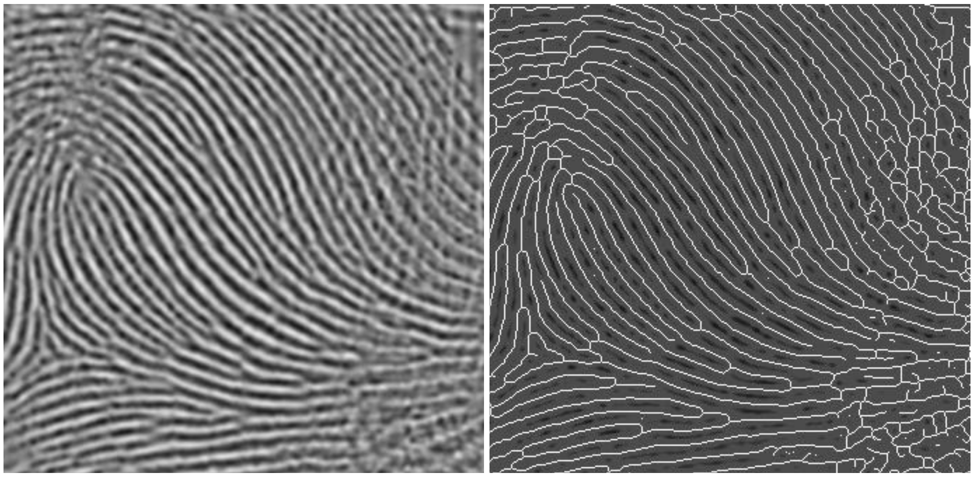

(a) (b)

Figure 5: A fingerprint image sample from NIST 14 sdb.

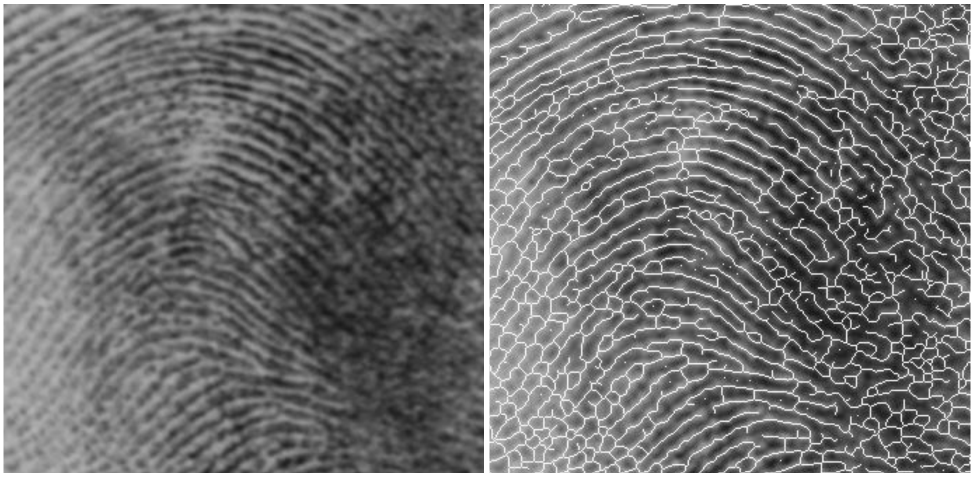

(a) (b)

Figure 6: A fingerprint image sample from NIST 4 sdb.