Songklanakarin J. Sci. Technol.
32 (3), 247-254, May - Jun. 2010

Songklanakarin Journal
of Science and Technology
http://www.sjst.psu.ac.th

Original Article

A Karnaugh-Map based fingerprint minutiae extraction method

Sunil Kumar Singla[1]* and Ajat Shatru Arora[2]

[1] *Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab, India*

[2] *Principal, DAV Institute of Engineering and Technology, Jalandhar, Punjab, India*

Received 4 March 2010; Accepted 14 May 2010

Abstract

Fingerprint is one of the most promising method among all the biometric techniques and has been used for the personal authentication for a long time because of its wide acceptance and reliability. Features (Minutiae) are extracted from the fingerprint in question and are compared with the features already stored in the database for authentication. Crossing number (CN) is the most commonly used minutiae extraction method for fingerprints. In this paper, a new Karnaugh-Map based fingerprint minutiae extraction method has been proposed and discussed. In the proposed algorithm the 8 neighbors of a pixel in a 3×3 window are arranged as 8 bits of a byte and corresponding hexadecimal (hex) value is calculated. These hex values are simplified using standard Karnaugh-Map (K-map) technique to obtain the minimized logical expression. Experiments conducted on the FVC2002/Db1_a database reveals that the developed method is better than the crossing number (CN) method.

Keywords: fingerprint, biometrics, minutiae extraction, crossing number, Karnaugh-Map

1. Introduction

Fingerprint recognition is one of the most mature biometric technologies used for human authentication and it is suitable for a large number of commercial and law enforcement applications. Fingerprints consist of raised friction ridges of skin separated by recessed valleys of skin (Jain *et al.*, 2006; Feng, 2008). A fingerprint is characterized by the curved formation of ridges on a finger. Human experts usually rely on local discontinuities in the ridge flow pattern called minutiae. According to an empirical study, two individuals will not have more than seven common minutiae (Maio and Maltoni, 1997). The set of minutiae are restricted into two types: Ridge endings and ridge bifurcations.

* Corresponding author.
Email address: sunilksingla2001@gmail.com

Ridge endings are the points where the ridge curve terminates and ridge bifurcations are the points where a ridge splits from a single path to two paths at a Y-junction as shown in Figure 1. The locations and angular orientations of the ridge endings and ridge bifurcations within the fingerprint uniquely characterize the fingerprint. The ridge ending and ridge bifurcation extraction algorithms known as minutiae extraction algorithms are of two types: Direct gray-scale extraction methods and binarization based methods.

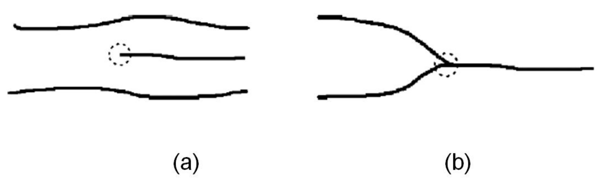


Figure 1. (a) Ridge ending (b) Ridge bifurcation.

Direct gray-scale extraction methods do not require binarization and thinning. Many direct gray-scale extraction methods are proposed in the literature. Leung *et al.* (1990) introduced a neural network-based approach in which the image is first transformed into a frequency domain where the filtering takes place and the resulting magnitude and phase signals constitute the input to the neural network composed of six sub-networks, each of which is responsible for detecting minutiae at a specific orientation. Finally, a classifier is employed to combine the intermediate responses. Maio and Maltoni (1997) proposed a technique based on a ridge line following algorithm that follows the image ridge lines and tries to locate the local maximum relative to a section orthogonal to the ridge direction until ridge ending or bifurcation occurs. Jiang *et al.* (2001) proposed a method similar to Maio and Maltoni (1997), except the ridge following step is dynamically adapted according to the changes in ridge contrast and bending level. In Liu *et al.* (2000), instead of tracking a single ridge, the algorithm simultaneously tracks a central ridge and two surrounding valleys. They search a central maximum and two adjacent minima in each section. In this method the ridge following step is dynamically adjusted according to the distances between lateral minima from the central maximum. Nilsson and Bigun (2001) proposed a method using linear symmetry (LS) properties computed by spatial filtering via separable Gaussian filters and Gaussian derivative filters. Minutiae are identified in the gray-scale image as points characterized by the lack of symmetry i.e. minutiae are local discontinuities of the LS vector field. In Leung *et al.* (1991), the method uses the three layer perceptron neural network to extract minutiae from the thinned binary image.

In binarization based methods, the most commonly employed method of minutiae extraction is the crossing number (CN) concept. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3×3 window of Figure 2. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighborhood. The CN for a ridge pixel *P* is given by (Arcelli *et al.*, 1984; Mehtre, 1993)

$$CN = 0.5\sum_{i=1}^{8}\left|P_i - P_{i+1}\right| \,,\ P_9 = P_1 \qquad (1)$$

If CN = 1 then the ridge pixel is a ridge ending, while if CN = 3 the ridge pixel is a ridge bifurcation otherwise it is a non-minutiae point. The crossing number method is not robust against the spikes (false minutiae) and tends to register spike as minutiae, which is required to be eliminated.

In this paper, a K-map based fingerprint minutiae extraction method, which is better than the crossing number method, has been proposed and in order to prepare an image for minutiae extraction: Filtering (Sobel filter followed by Gaussian filter), segmentation (by calculating the variance and mean), regional average thresholding (Eminoglu *et al.*, 1997), and thinning (Wang and Wu, 2004) as pre-processing steps have been used.

2. Proposed method

In the proposed method ridge ending and ridge bifurcation (minutiae) are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3×3 window of Figure 2.

2.1 Ridge Ending

From the truth table given in Table 1, a pixel in an image is a ridge ending if
{(Sum of 3 ×3 window is 2) + [(Sum of 3 × 3 window is equal to 3) & (P_1&P_2) + (P_2&P_3) + (P_3&P_4) + (P_4&P_5) + (P_5&P_6) + (P_6&P_7) + (P_7&P_8) + (P_1&P_8) }]}= 1

There are total 16 genuine ridge ending conditions and are shown in Table 1.

The minimized logical expression solved by K-map of all these 16 conditions is given by the following logical expression:

$(\sim P_1 \& \sim P_2 \& \sim P_3 \& P_5 \& \sim P_6 \& \sim P_7 \& \sim P_8) + (\sim P_1 \& \sim P_2 \& \sim P_3 \& \sim P_4 \& P_6 \& \sim P_7 \& \sim P_8) + (\sim P_1 \& \sim P_2 \& \sim P_3 \& \sim P_4 \& \sim P_5 \& P_7 \& \sim P_8) + (\sim P_1 \& \sim P_2 \& \sim P_3 \& \sim P_4 \& \sim P_5 \& \sim P_6 \& P_8) + (P_1 \& \sim P_2 \& \sim P_3 \& \sim P_4 \& \sim P_5 \& \sim P_6 \& \sim P_7) + (P_2 \& \sim P_3 \& \sim P_4 \& \sim P_5 \& \sim P_6 \& \sim P_7 \& \sim P_8) + (\sim P_1 \& P_3 \& \sim P_4 \& \sim P_5 \& \sim P_6 \& \sim P_7 \& \sim P_8) + (\sim P_1 \& \sim P_2 \& P_4 \& \sim P_5 \& \sim P_6 \& \sim P_7 \& \sim P_8) = 1$

where ~ stands for Logical NOT operation, + stands for Logical OR operation, & stands for Logical AND operation.

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figure 2. 3×3 operation window.

Figure 3. Ridge bifurcation example.

Table 1. Truth table of ridge ending.

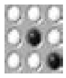
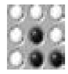
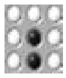
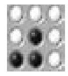
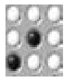
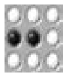
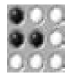
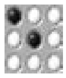
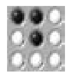
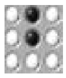
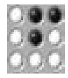
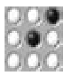
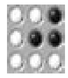
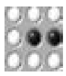
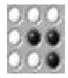
P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
0	0	0	0	0	0	0	1		0	0	0	0	0	0	1	1	
0	0	0	0	0	0	1	0		0	0	0	0	0	1	1	0	
0	0	0	0	0	1	0	0		0	0	0	0	1	1	0	0	
0	0	0	0	1	0	0	0		0	0	0	1	1	0	0	0	
0	0	0	1	0	0	0	0		0	0	1	1	0	0	0	0	
0	0	1	0	0	0	0	0		0	1	1	0	0	0	0	0	
0	1	0	0	0	0	0	0		1	1	0	0	0	0	0	0	
1	0	0	0	0	0	0	1		1	0	0	0	0	0	0	0	

Table 2. Truth table of ridge bifurcation.

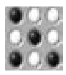
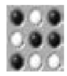
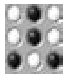
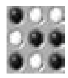
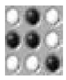
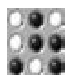
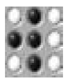
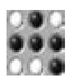
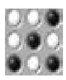
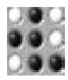
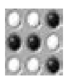
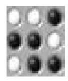
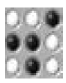
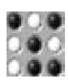
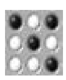
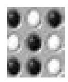
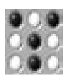
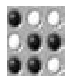
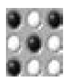
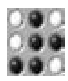
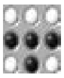
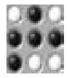
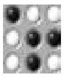
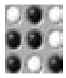
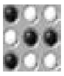
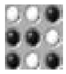
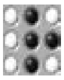
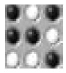
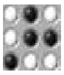
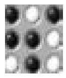
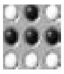
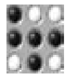
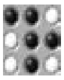
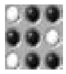
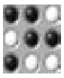
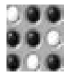
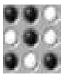
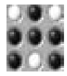
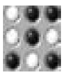
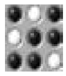
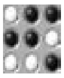
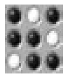
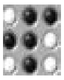
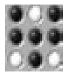
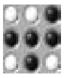
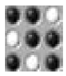
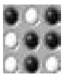
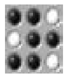
P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
0	0	0	1	0	1	0	1		1	1	0	1	0	1	0	0	
0	0	1	0	0	1	0	1		1	0	0	1	0	1	0	1	
0	0	1	0	1	0	0	1		1	0	1	0	0	1	0	1	
0	0	1	0	1	0	1	0		1	0	1	0	1	0	0	1	
0	1	0	0	0	1	0	1		0	0	1	0	1	0	1	1	
0	1	0	0	1	0	0	1		0	1	0	0	1	0	1	1	
0	1	0	0	1	0	1	0		0	1	0	1	0	0	1	1	
0	1	0	1	0	0	0	1		0	1	0	1	0	1	1	0	
0	1	0	1	0	0	1	0		1	0	0	1	0	1	1	0	
0	1	0	1	0	1	0	0		1	0	1	0	0	1	1	0	

Table 2. (Continued)

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
1	0	0	0	1	0	1	0		1	0	1	0	1	1	0	0	
1	0	0	1	0	0	1	0		0	0	1	0	1	1	0	1	
1	0	0	1	0	1	0	0		0	1	0	0	1	1	0	1	
1	0	1	0	0	0	1	0		0	1	0	1	1	0	0	1	
1	0	1	0	0	1	0	0		0	1	0	1	1	0	1	0	
1	0	1	0	1	0	0	0		1	0	0	1	1	0	1	0	
1	0	1	1	0	0	1	0		0	1	1	0	1	0	1	1	
1	0	1	1	0	1	0	0		0	1	1	0	1	1	0	1	
0	0	1	1	0	1	0	1		1	0	1	0	1	1	0	1	
0	1	1	0	0	1	0	1		1	1	0	1	0	1	1	0	
0	1	1	0	1	0	0	1		0	1	0	1	1	0	1	1	
0	1	1	0	1	0	1	0		1	1	0	1	1	0	1	0	
1	1	0	0	1	0	1	0		1	0	1	1	0	1	0	1	
1	1	0	1	0	0	1	0		1	0	1	1	0	1	1	0	

2.2 Ridge Bifurcation

Each entry in the Table 2 represents the condition for ridge bifurcation e.g. for window of Figure 2 if, pixel values are $P_1 = 0$, $P_2 = 1$, $P_3 = 0$, $P_4 = 1$, $P_5 = 0$, $P_6 = 1$, $P_7 = 0$, $P_8 = 0$ then the resulted ridge bifurcation is shown in Figure 3. From the truth table given in Table 2, a pixel in an image is a ridge bifurcation if

(Sum of 3 3 window is either 4 or 5 or 6) & (Any of the condition shown in Table 2) =1

In all, 48 such cases are formed for ridge bifurcation as given in Table 2. The minimized logical expression solved by K-map of all 48 bifurcation conditions is given by the following expression:

$(\sim P_1\&P_2\&\sim P_4\&P_5\&\sim P_6\&P_8)+(P_2\&\sim P_3\&P_5\&\sim P_6\&P_7\&\sim P_8)+(\sim P_1\&P_3\&\sim P_4\&P_5\&\sim P_6\&P_7)+(P_1\&\sim P_2\&P_3\&\sim P_4\&P_6\&\sim P_7)+(\sim P_1\&P_2\&\sim P_4\&P_6\&\sim P_7\&P_8)+(\sim P_1\&P_3\&\sim P_4\&P_6\&\sim P_7\&P_8)+(\sim P_2\&P_4\&\sim P_5\&P_6\&\sim P_7\&P_8)+(P_1\&\sim P_2\&P_4\&\sim P_5\&P_6\&\sim P_8)+(P_2\&\sim P_3\&P_4\&\sim P_5\&P_6\&\sim P_8)+(P_1\&\sim P_2\&P_4\&\sim P_5\&P_7\&\sim P_8)+(P_1\&\sim P_2\&P_3\&\sim P_5\&P_7\&\sim P_8)+(P_2\&\sim P_3\&P_4\&\sim P_5\&P_7\&\sim P_8)+(\sim P_1\&P_2\&\sim P_3\&P_4\&\sim P_6\&P_8)+(P_1\&\sim P_2\&P_3\&\sim P_4\&P_5\&\sim P_6\&\sim P_7)+(\sim P_2\&P_3\&\sim P_4\&P_5\&\sim P_6\&\sim P_7\&P_8)+(P_1\&\sim P_2\&\sim P_3\&P_5\&\sim P_6\&P_7\&\sim P_8)=1$

All the conditions given in Table 1 and Table 2 are solved using the K-map to obtain the combined minimized logical expression. The resulting K-map is shown in Figure 4.

	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
0000	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1
0001	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0011	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
0010	1	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0
0110	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	1
0111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0101	1	1	0	1	0	1	0	1	0	0	0	1	0	0	1	1
0100	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1
1100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1101	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
1111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1010	0	0	0	1	1	0	1	1	1	1	0	0	0	0	1	1
1011	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0
1001	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0
1000	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Figure 4. K-map of combined minutiae extraction rules.

The combined minimized logical expression for minutiae (ridge ending and bifurcation) extraction is given by

$$
\begin{aligned}
X = {} & (P_2\&\sim P_3\&P_5\&\sim P_6\&\sim P_7\&\sim P_8)+(\sim P_1\&P_2\&\sim P_4\&P_5\&\sim P_6\&\sim P_7)+(\sim P_2\&P_3\&\sim P_4\&P_6\&\sim P_7\&P_8)\\
& +(\sim P_2\&P_4\&\sim P_5\&P_6\&\sim P_7\&P_8)+(P_1\&\sim P_2\&P_4\&\sim P_5\&P_6\&\sim P_8)+(P_1\&\sim P_2\&P_3\&\sim P_5\&P_6\&\sim P_8)\\
& +(P_1\&\sim P_2\&P_3\&\sim P_5\&P_7\&\sim P_8)+(\sim P_1\&P_2\&\sim P_4\&\sim P_6\&P_7\&P_8)+(P_2\&\sim P_3\&P_4\&\sim P_5\&\sim P_7\&\sim P_8)\\
& +(P_2\&\sim P_3\&P_4\&\sim P_5\&\sim P_6\&\sim P_8)+(\sim P_1\&P_2\&\sim P_3\&P_4\&\sim P_6\&\sim P_7)+(\sim P_2\&P_3\&\sim P_4\&P_5\&\sim P_6\&\sim P_7\&P_8)\\
& +(P_1\&\sim P_2\&\sim P_3\&P_5\&\sim P_6\&P_7\&\sim P_8)+(\sim P_1\&\sim P_2\&P_3\&\sim P_4\&P_5\&\sim P_6\&P_7)+(\sim P_2\&P_3\&P_4\&P_5\&P_6\&P_7\&\sim P_8)\\
& +(P_1\&\sim P_2\&P_3\&\sim P_4\&P_5\&\sim P_7\&\sim P_8)+(\sim P_1\&\sim P_2\&\sim P_3\&\sim P_4\&P_5\&\sim P_7\&\sim P_8)+(\sim P_1\&P_2\&\sim P_3\&\sim P_4\&P_6\&P_7\&P_8)\\
& +(\sim P_1\&P_2\&\sim P_3\&\sim P_5\&P_6\&P_7\&P_8)+(\sim P_1\&P_2\&\sim P_4\&\sim P_5\&P_6\&P_7\&P_8)+(\sim P_1\&\sim P_2\&\sim P_3\&\sim P_4\&\sim P_5\&P_6\&\sim P_8)\\
& +(P_1\&\sim P_2\&\sim P_3\&P_4\&\sim P_6\&P_7\&\sim P_8)+(\sim P_1\&\sim P_2\&\sim P_3\&\sim P_4\&\sim P_5\&\sim P_6\&P_7)+(\sim P_1\&\sim P_2\&\sim P_3\&\sim P_4\&\sim P_5\&\sim P_6\&P_8)\\
& +(P_1\&\sim P_2\&\sim P_3\&\sim P_4\&\sim P_5\&\sim P_6\&\sim P_7)+(\sim P_1\&\sim P_2\&P_3\&\sim P_5\&\sim P_6\&\sim P_7\&\sim P_8)+(\sim P_1\&\sim P_2\&\sim P_3\&P_4\&\sim P_6\&\sim P_7\&\sim P_8)\\
& +(\sim P_1\&P_2\&\sim P_3\&P_4\&P_5\&P_6\&P_7\&\sim P_8);
\end{aligned}
$$

$$
\text{If } X = \begin{cases} 1 & \text{P is a Minutiae Point} \\ 0 & \text{Not a Minutiae Point} \end{cases}
$$

where P is the pixel in question.

3. Results and Discussions

In this paper, an improved feature (minutiae) extraction algorithm using minimum logical expression has been discussed. The experiments have been conducted on the 800 images in the FVC2002/Db1_a database, and a comparison of crossing number and proposed method for 20 images is shown in Table 3, which indicates that the proposed method of minutiae extraction is far better than the crossing number method. Table 3(a) shows the comparison of minutiae extracted by the two techniques, while Table 3(b) gives the comparison of ridge ending and ridge bifurcations extracted through the two methods. Table 3(b) also shows the coordinates (considering the coordinates of top left corner pixel as {1,1}) of the false ridge bifurcation points eliminated by the proposed method.

The comparison of two methods reflects that the proposed method eliminates almost 10% false minutiae in the extraction stage, which remains with the crossing number method. The reasons for the improvement are:

- The minutiae owing to the spike structure are eliminated because for a spike to occur, one of the conditions shown in Figure 5 must be present i.e.

$$[(P_1 \& P_2 \& P_8) + (P_2 \& P_3 \& P_4) + (P_4 \& P_5 \& P_6) + (P_6 \& P_7 \& P_8)] = 1$$

For the crossing number method these are ridge ending conditions but in the proposed method these conditions are eliminated.

- As shown in Figure 6, If in a 3×3 window four or less consecutive pixels are zero, ridge ending conditions will not occur in actual but crossing number method consider these conditions as ridge endings. All these conditions are also eliminated by the proposed method.

Figure 5. Conditions for spike.

Table 3a. Comparison of the minutiae extracted by the two techniques.

Sr. No.	Image	CN Method	Proposed	Difference	% Difference
1	3_7	1032	937	95	9.21
2	3_4	743	666	77	10.36
3	4_6	552	514	38	6.88
4	4_4	425	399	26	6.12
5	5_2	621	581	40	6.44
6	5_6	1103	1070	33	2.99
7	6_7	859	813	46	5.36
8	6_8	518	502	16	3.09
9	7_5	1228	1173	55	4.48
10	7_7	1053	1039	14	1.33
11	11_5	975	902	73	7.49
12	11_6	728	665	63	8.65
13	12_6	1471	1368	103	7.00
14	12_4	1685	1642	43	2.55
15	13_1	571	523	48	8.41
16	13_2	703	655	48	6.83
17	14_8	820	790	30	3.66
18	14_6	670	655	15	2.24
19	15_7	739	659	80	10.83
20	15_8	781	739	42	5.38

Table 3b. Comparison of ridge ending and ridge bifurcation extracted by the two techniques.

Sr. no.	Image	CN Method		Proposed		Difference		Pixels coordinates(X/Y) where false bifur detected
		Ending	Bifur	Ending	Bifur	Ending	Bifur	
1	3_7	883	149	790	147	93	2	(172/150), (352/228)
2	3_4	688	55	611	55	77	0	
3	4_6	487	65	451	63	36	2	(265/282), (265/284)
4	4_4	386	39	360	39	26	0	
5	5_2	528	93	490	91	38	2	(325/329), (231/322)
6	5_6	912	191	880	190	32	1	(178/201)
7	6_7	737	122	691	122	46	0	
8	6_8	445	73	431	71	24	2	(307/144), (307/146)
9	7_5	1017	211	963	210	54	0	
10	7_7	985	68	972	67	13	1	(223/86)
11	11_5	917	58	845	57	72	1	(155/298)
12	11_6	668	60	605	60	63	0	
13	12_6	1199	272	1097	271	102	0	
14	12_4	1507	178	1470	172	37	6	(66/180), (178/220), (217/173), (275/230), (348/152), (360/285)
15	13_1	515	56	467	56	48	0	
16	13_2	640	63	592	63	48	0	
17	14_8	740	80	712	78	28	1	(278/292), (278/294)
18	14_6	563	107	548	107	15	0	
19	15_7	640	99	561	98	79	1	(186/274)
20	15_8	636	145	594	145	42	0	

- The false ridge bifurcation conditions shown in Figure 7 are also eliminated by the proposed method, which are considered as bifurcations by the CN method.

Crossing number and the proposed method of minutiae extraction are developed with the Laboratory Virtual Instruments Engineering Workbench (LabVIEW), version 6i.

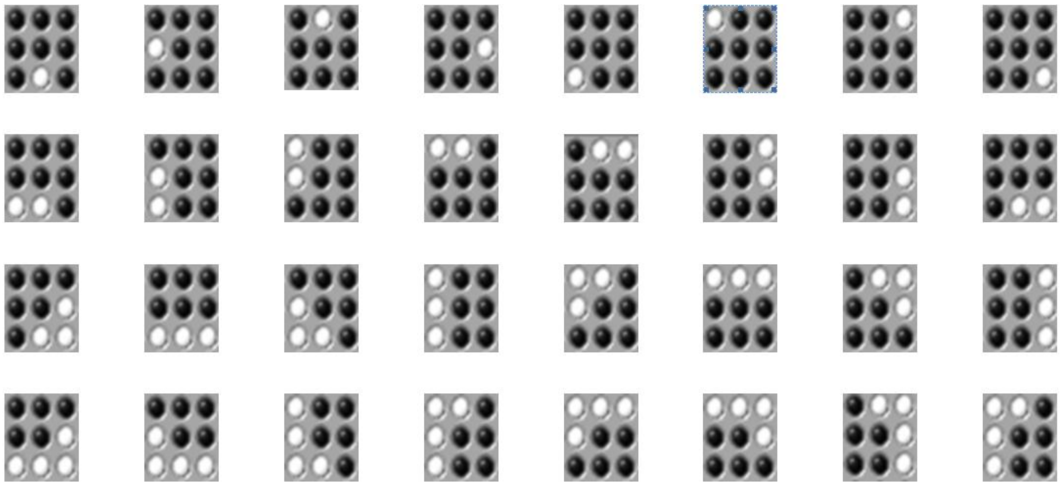


Figure 6. False ridge ending conditions considered by CN method.

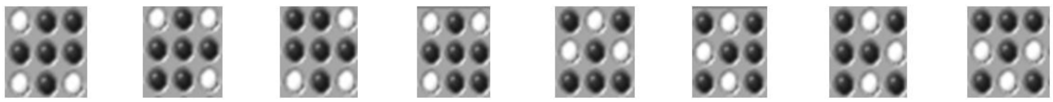


Figure 7. False ridge bifurcation conditions considered by CN method.

The expanded portions of thinned fingerprint images from the FVC 2002/Db1_a database are shown in Figure 8, in which the marked 3×3 windows in red colour shows some of false minutiae points, to which the CN method counts as valid minutiae (ridge endings or bifurcations) points. These points need to be eliminated to obtain the better efficiency of the method. The proposed algorithm helps to eliminate these points automatically.

4. Conclusion

Feature extraction is a very important step in a human authentication system. The minutiae based identification systems have been popular because of their accuracy and less complexity. The crossing number method is one of the most widely used binarization based method for minutiae extraction but is not robust against the spikes and tends to register spike as minutiae. An improved method has been proposed for the minutiae extraction, which uses genuine cases only. All these cases are solved to a minimized logical expression using a well known minimization technique of K-map. The proposed method eliminates up to 10% false minutiae in the extraction stage, which remains with the crossing number method. It has also been observed that while there is significant improvement in identification of ridge endings, only a few false ridge bifurcations are eliminated. These results are in tune with the difference in number of false ridge ending and bifurcation cases identified by the proposed method.

References

Arcelli, C. and Baja, G.S.D. 1984. A width independent fast thinning algorithm. The Institute of Electrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence. 4(7), 463-474.

Emiroglu, I. and Khan, M.B.A. 1997. Preprocessing of fingerprint images. Proceedings of the European Conference on Security and Detection, Commomwealth Institute, London, April 28-30, 1997, 147-151.

Feng, J. 2008. Combining minutiae descriptor for fingerprint matching. The Journal of the Pattern Recognition Society. 41, 342-352.

Jain, A.K., Ross, A. and Pankanti, S. 2006. Biometrics: A Tool for Information Security. The Institute of Electrical and Electronics Engineers (IEEE) Transactions on Information Forensics and Security. 1(2), 125-142.

Jiang, X., Yau, W.Y. and Ser, W. 2001. Detecting the fingerprint minutiae by adaptive tracking the grey level ridge. Pattern Recognition. 34(5), 999-1013.

Leung, M., Engeler, W. and Frank, P. 1990. Fingerprint image processing using neural network. Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Region Conference on Computer and Communication Systems, Hong Kong, Sep 24-27, 1990, 582-586.

Leung, W. F., Leung, S.H., Law, W.H. and Luk, A. 1991. Fingerprint recognition using neural network. Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Workshop on Neural Network for Signal Processing, Princeton, New Jersey, USA, Sept. 29 - Oct. 2, 1991, 226-235.

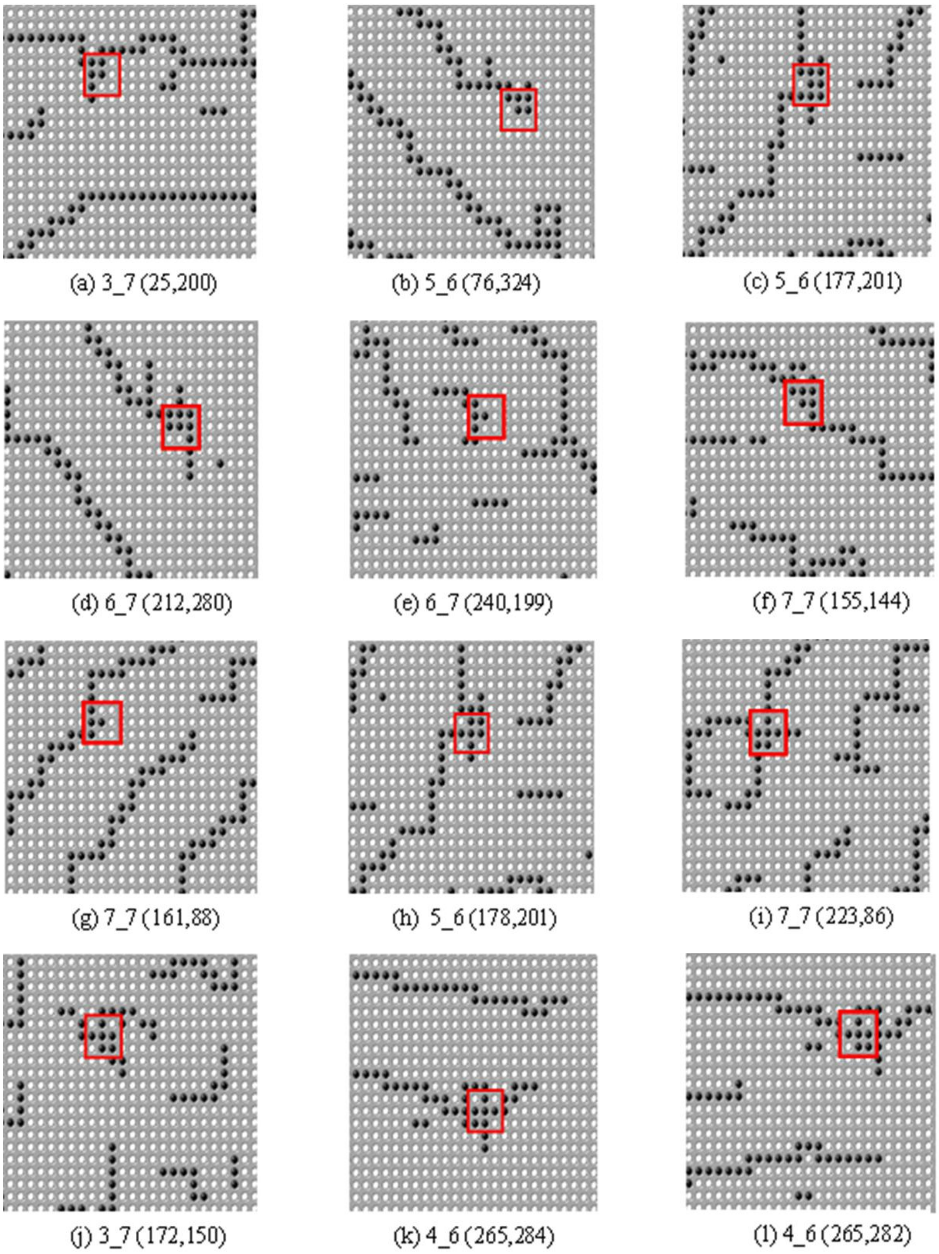


Figure 8. Expanded versions of the different parts of the thinned images. Rectangular window shows the false minutiae (a,b,c,d,e,f,g ridge endings, while h,i,j,k,l ridge bifurcations) rejected by proposed algorithm. (Caption shows FVC 2002/Db1_a database Fig. no. followed by coordinates of the pixel in question).

Liu, J., Huang, Z. and Chan, K. 2000. Direct minutiae extraction from gray level fingerprint image by relationship examination. Proceedings of the International Conference on Image Processing, Vancouver, BC, Canada, September 10-13, *2000,* 427-430.

Maio, D. and Maltoni, D. 1997. Direct Gray- Scale Minutiae Detection in fingerprints. The Institute of Electrical and Electronics Engineers (IEEE) Transactions on Pattern Analysis and Machine Intelligence. 19(1), 27-40.

Mehtre, B.M. 1993. Fingerprint image analysis for automatic identification. Machine Vision and Applications. 6(2), 124-139.

Nilsson, K. and Bigun, J. 2001. Using linear symmetry features as a preprocessing step for fingerprint images. Proceedings of the International Conference on Audio and Video Based Biometric Person Authentication, Halmstad, Sweden, June 6-8, *2001,* 247-252.

Wang, C. and Wu, K. T. 2004. Design of a pixel array circuit for thinning process. Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Symposium on Circuit and Systems, Sheraton Vancouver, BC, Canada, May 23 - 26, 2004, 89-92.