IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Boundary-Polygons for Minutiae based Fingerprint Recognition

Kusha Maharshi[1], Prashant Sahai Saxena[2]

Student, Class 12, Neerja Modi School, Jaipur[1]

Professor, School of Computer and System Sciences, Jaipur National University, Jaipur[2]

Abstract: Over the years, researchers have developed numerous algorithms for pattern recognition, each one being either a novel approach or a significant improvement over the previous work. Also, each new concept for identifying patterns finds uses in many more fields than what it was initially moulded for. Pattern recognition has found indispensable use in human identification based on unique biometric features. A biometric widely used for identification is the fingerprint. In this paper, we propose and explain a new method for minutiae-point-based pattern recognition for fingerprint identification. Comparison between the angles and edge lengths of polygons formed from the minutiae extracted from a fingerprint image is used as the criterion for matching.

Keywords: Point pattern recognition, boundary polygon algorithm, fingerprint recognition, minutiae based matching.

I. INTRODUCTION

Pattern recognition is storing, identifying and matching patterns computationally. A major challenge of the field is to effectively store the information gathered from the extracted features using less memory while also ensuring that the obtained data is enough for a valid matching decision.

Computational complexity, time taken to reach the desired output and invariance to rotation and reasonable distortion are also significant factors to keep in mind while developing effective algorithms for pattern recognition.

Pattern recognition has applications in fingerprint identification, speech recognition, text classification, image processing, etc. A particular way of recognising patterns is using special points to uniquely or selectively identify patterns in the database in order to match with the user's input.

It is essential to make the matching of these 'points' as concise and precise as possible. In this paper, we discuss an approach to point based pattern recognition by applying it to fingerprint recognition.

Fingerprint recognition is widely used as a reliable biometric feature for identification. We have extracted the minutiae points from fingerprint images and used them to form polygons.

The angles and lengths of the edges of these polygons have been used to match different impressions of the same fingerprint.

II. LITERATURE REVIEW

Researchers have proposed various approaches for matching the input and template fingerprints:

1) Jain et al [1] have proposed an approach in which a region of interest with respect to a reference point in the fingerprint image is tessellated. The information about local features is obtained from the sectors in the tessellation. The tessellation is sequenced to provide global information, and the obtained data is stored as FingerCode. Matching is done by calculating the Euclidean distance between the FingerCodes.
2) Nalini K Ratha [2] has used graphs for representing minutiae points and further working on their comparative differences for matching. Many more algorithms using graphs for analysis and matching have been proposed.
3) Fingerprint identification is based on triangular matching in the method devised by Kovacs-Vajna [3]. This is followed by dynamic time warping to finally match the concerned fingerprint images.
4) Saleh et al [4] consider 10-pixel wide regions around the core point. The number of minutiae in each region is used as a matching criterion.
5) Jiang and Yau [5] have used the local and global structures for determining the uniqueness of a fingerprint. The relation between a feature point and its neighbouring minutiae combined with the generic pattern of the fingerprint are used to increase the reliability of the algorithm.
6) X. Tan et al [6] view the matching problem as an algorithm to find the "optimised transformation" between template's minutiae and the input minutiae.

III.METHODOLOGY

The entire process from building the fingerprint database to verifying an input fingerprint image is conducted in the following stages:

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

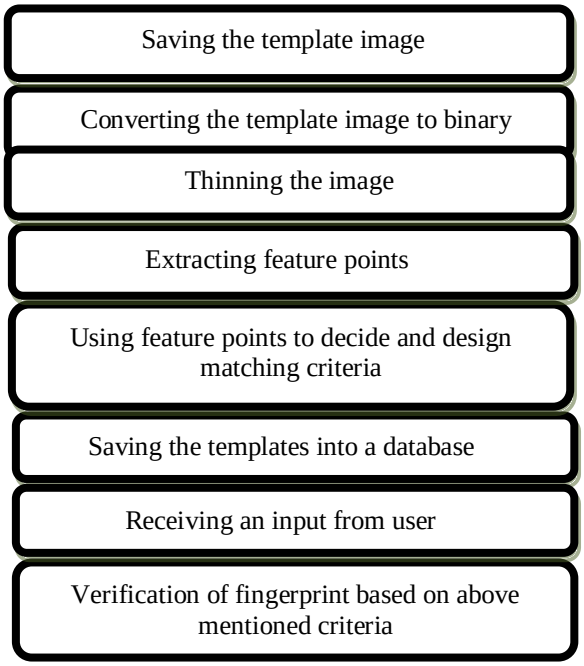

Fig. 1 Methodology

A. Pre-processing

For saving a person's fingerprint image in the database, the person is required to provide 6-7 impressions of his or her fingerprint in order to ensure storage of all feature points. This is known as training. After information from all the impressions is available, these images are converted to binary and all information is stored within arrays of zeros and ones. The binary image is then 'thinned' (Figure 2), which is, essentially, eroding the image until it is reduced to a skeleton of the original image. Then, a noise-reduction filter is in order. The spurious, isolated points are removed to ensure valid feature detection later in the ladder of fingerprint matching. The final image is the one we can work upon further.

Fig.2 A section of a fingerprint's thinned image

B. Feature Extraction

First of all, it is important to know what exactly these features are. As is well established by now, each person has a unique fingerprint. A fingerprint is identified based on points called ridges and bifurcations. Visually and globally, there are various broad classifications under which fingerprint patterns can be classified. These include right loop, left loop, arch, tented arch, whorl [7]. Computationally, fingerprints are identified based on the relative arrangement of ridges and bifurcations. Ridges and bifurcations are the binary antipodes of each other. A point is called a ridge if it denotes an end to a continuous string of black pixels. A bifurcation, on the other hand, denotes the beginning of two new streams of black pixels.

If these black and white pixels are swapped, an initial bifurcation would be a new ridge. In order to detect these features (referred to as minutiae), an algorithm is employed to find the points isolated at ridge endings. We use the crossing number method [8] on a thinned image for its computational simplicity and accuracy, if tweaked a little.

One may imagine a sliding 3x3 grid which passes over each black pixel in the image. The eight neighbouring pixels of each pixel are taken into account. If a pixel has only one black, neighbouring pixel, it is a ridge. But one must consider the limitations to this. In case of poor quality input images, the corresponding thinned images may contain several short branches which may vary in each impression due to the poor scanning quality. Also, these branches will unnecessarily increase the number of minutiae used for recognition. So, for each ridge point, its neighbour black pixel must have two neighbouring black pixels (including the ridge point). The next new neighbour must also satisfy the same condition and this criterion should loop until a predefined threshold value for number of black pixels in a ridge line is reached.

A new array of binary image points is made, where the ones are replaced with zeros and vice versa. The same method as the one employed for ridge identification can now be used for bifurcation identification. The bifurcations (Figure 3) and ridges (Figure 4) are stored in terms of their positions in the initial binary image array in newly defined arrays.

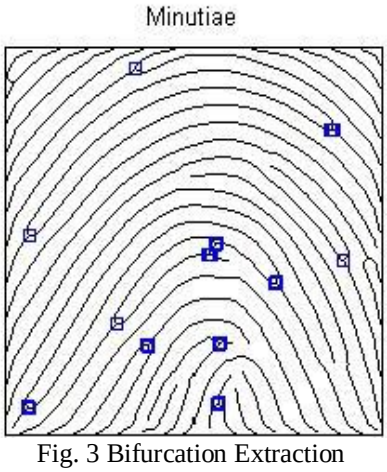

Fig. 3 Bifurcation Extraction

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

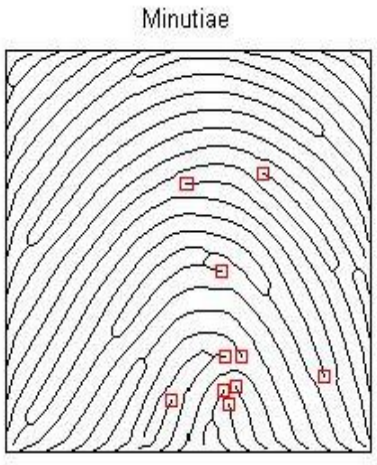

Fig. 4 Ridge Extraction

IV. MATCHING

A. Using minutiae for matching criteria

Let us consider the array of bifurcation points (the same method can be applied to ridges). This array is converted into an array of x and y coordinates and plotted (Figure 5).

One may now imagine these points to have heights in the third dimension. If a rubber band is stretched around this set of points it will touch some points lying on the outermost boundary. The points on this outermost boundary are extracted and stored into an array. Now, the outer most boundary points (which form a polygon when joined) of the remaining (i.e. the initial set of points-the points in the first outermost boundary) are extracted and stored (Figure 6, 7, 8, 9). This is continued until the number of remaining points is greater than 1 (a line will be formed in case of two points).

Each time the algorithm extracts and stores boundary polygons, it also sorts them in clockwise order. Additionally, it calculates the angles formed between the adjacent edges and the length of these edges of the polygons and stores them.

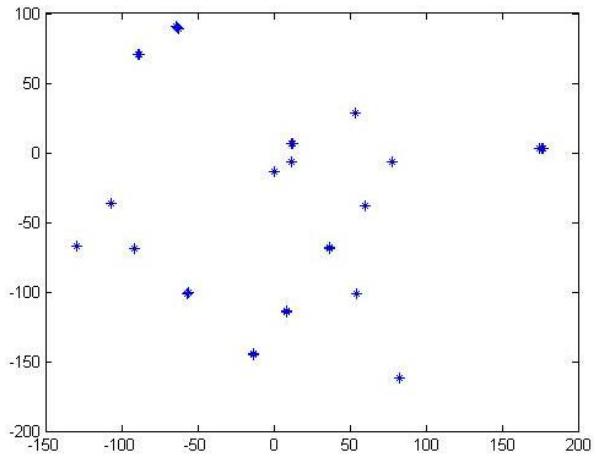

Fig. 5 Demonstration of boundary polygon algorithm

Also the points in a polygon at which the minimum angles are formed are found. Such points from all polygons are then used to measure the relative distances between them. This "fixes" the pattern of polygons that each fingerprint image is. The stored information is then matched, based on a threshold value of similar features, with that obtained from input images to verify fingerprints.

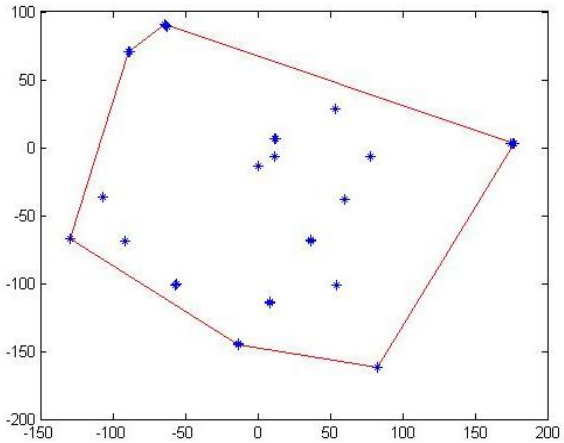

Fig. 6 First boundary polygon

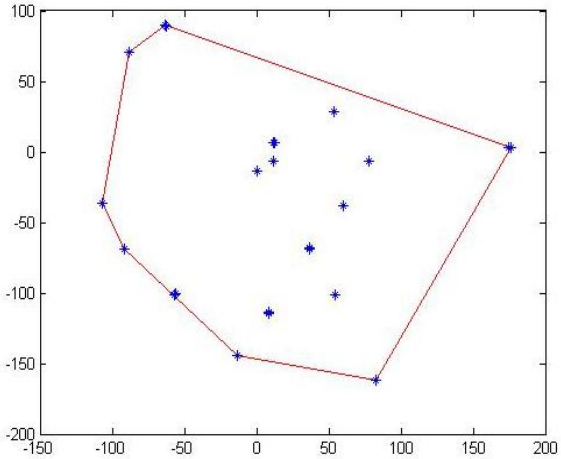

Fig. 7 Second boundary polygon

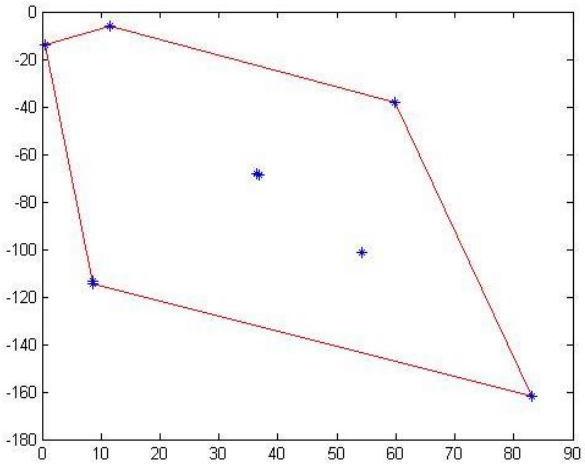

Fig. 8 Second last boundary polygon

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

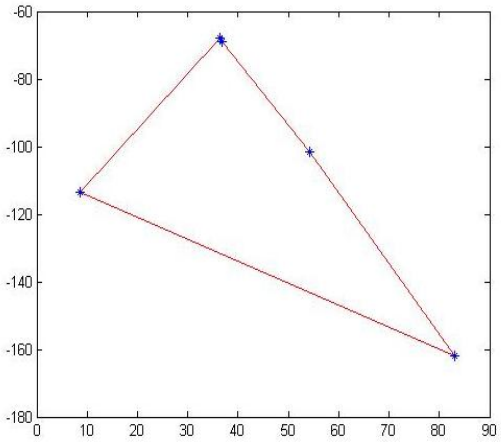

Fig. 9 Last boundary polygon

B. Calculating the angles and distances

It is imperative that we get the correct interior angles of the polygons in order to obtain the right results. Simply applying a loop on every vertex of a polygon will produce negative values (or values with a difference of 180 degrees) of angles in some cases. Therefore, some conditions are applied which take into account the relative positions of the previous and next vertex of the polygon in clockwise direction. Using the inverse tangents of all the edges and modifying them by taking their negative or adding/subtracting 180 degrees, we accurately calculate the interior angles. The distances between these points are calculated using the Euclidean distance formula.

C. Addressing the problem

Each time the system receives an impression from the same person, we can expect some variations. These variations include different orientations, generation of spurious minutiae, absence of peripheral minutiae and gross reduction in the area of image received. The reason behind using relative angles and some distances is that the orientation of fingerprint should not affect the verification decision. The spurious minutiae are accounted for in the threshold set for impressions from the same person. But all of the stored minutiae may not be present in the input image. Some may not be in the field of view of the scanner. Generally, the number of missing points is low enough to obtain a series of partially matching polygons, which is again accounted for in the threshold, but for the sake of accuracy, let us consider the case whereby the extra or missing minutiae bring about the formation of a new polygon or absence of one. In other words, the series of polygons will be significantly disrupted due to this change. To tackle this problem, we use a smaller set of minutiae points which fall within a circular region, with a given radius, centred at a selected point. This point is the core point which, once extracted, serves as a detectable point common to both - the template image and the input image .The selected region of points is matched with a region with the same radius from the template image, centred at the core of template points. Also, the minimum available distance from the core point in the input image is calculated. So, at the time of matching, only the minutiae points falling within a circular region of radius equal to that minimum distance are taken into consideration for matching.

Last but not the least, sometimes only a small part of a fingerprint may be available such that the number of minutiae is not enough to satisfy the threshold. This is, for example, very crucial in crime scene investigation, whereby the concerned person makes no effort to provide a reasonable area of finger for matching the fingerprint. In this case, a reverse matching of an embedded pattern must be employed to narrow down possible matching candidates.

D. Embedded fingerprint patterns

If we have a small set of minutiae points, first the point closest to the mean point is found. Now the boundary polygons formed from the remaining points around this polygon are formed and the angles are stored as earlier. The radius of this available region (r) is also stored (centre of the circle is the centroid of the available minutiae). Now, all points (converted from binary to Cartesian coordinates) in all available template images are tested. Each point is considered the centre of a circular region of radius ‘r’. All minutiae points falling within this radius are extracted and the boundary polygon algorithm is applied. So, for each pixel in an image, the surrounding region is checked. This way the template image(s) containing the embedded pattern is found.

E. Illustration

The extracted angles are stored in consecutive arrays. The total number of arrays obviously varies for different fingerprints. Here is a comparison of angles retrieved from two impressions from the same fingerprint (FVC 2000 Database2 B):

TABLE I ANGLES FROM OUTERMOST RING OF RIDGES

Impression_1 (Outermost ridge ring)	**Impression_2 (Outermost ridge ring)**
117.7137	118.7234
112.1380	112.7113
77.1312	70.2523
53.0171	58.3130

TABLE III ANGLES FROM INNERMOST POLYGON OF BIFURCATIONS

Impression_1 (Innermost bifurcation polygon)	**Impression_2 (Innermost bifurcation polygon)**
137.2297	138.7586
84.4992	83.6440
115.782	115.796
159.2118	158.5523
134.2559	133.8736
89.0214	89.3755

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

TABLE IIIII EDGE LENGTHS FROM OUTERMOST POLYGON OF RIDGES

Impression_1 (Outermost ridge ring)	**Impression_2 (Outermost ridge ring)**
24.5153	19.1050
73.4983	83.2406
97.4987	96.1769
94.6414	93.2309

TABLE IVV EDGE LENGTHS FROM INNERMOST POLYGON OF BIFURCATIONS

Impression_1 (Innermost bifurcation polygon)	**Impression_2 (Innermost bifurcation polygon)**
107.2054	108.1665
65.3911	66.6408
73.0616	74.0000
28.4605	30.0832
68.8767	68.1542
59.5063	59.2368

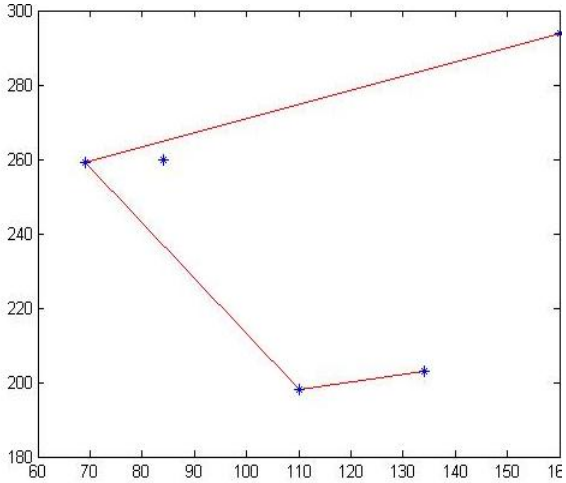

Fig. 10 Impression_1 bifurcation innermost ring

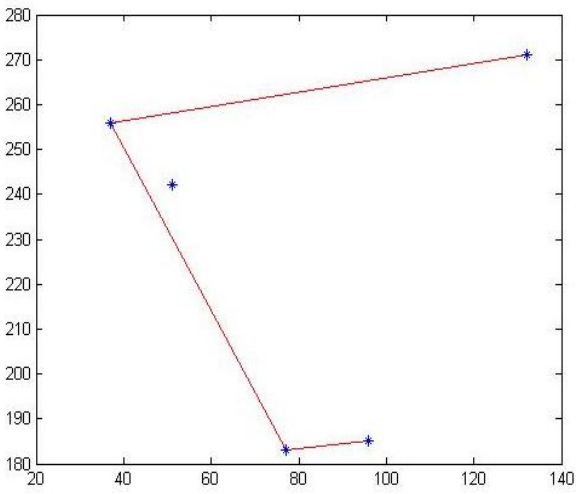

Fig. 11 Impression_2 bifurcation innermost ring

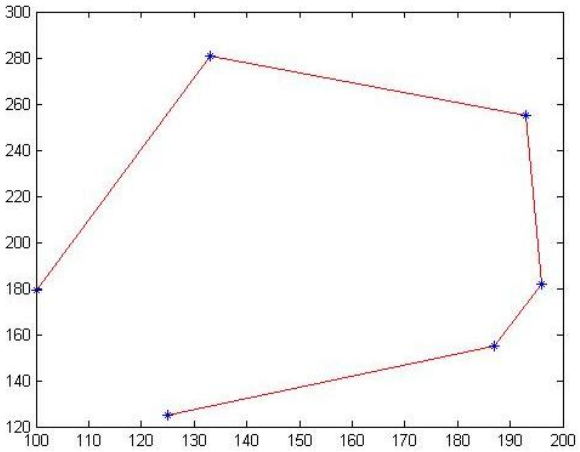

Fig. 12 Impression_1 ridge outermost ring

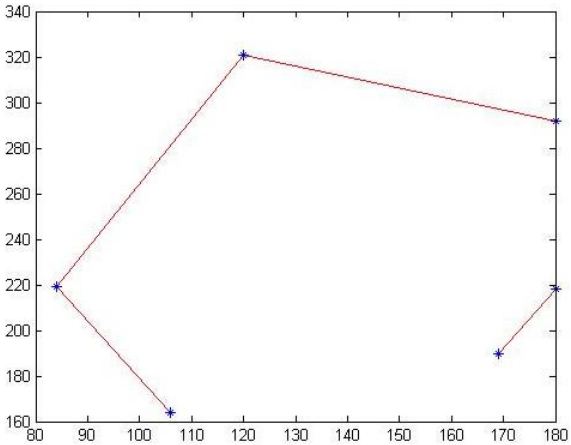

Fig. 13 Impression_2 ridge outermost ring

V. RESULTS

The above mentioned method was implemented using MATLAB. On FVC 2000 database 2 B, the following results were obtained:

TABLE V RESULTS

Accuracy	96.67%
Average time per template per match (with images)	01.84 seconds
Average time per template per match (without images)	00.33 seconds
Average data size per image	27.73 kilobytes

VI. CONCLUSION AND FUTURE SCOPE

The matching threshold can be refined in order to resolve the conflict between accepting heavily distorted images and similar impressions from different fingerprints. Also, the proposed technique for fingerprint matching can be applied to other fields as well.

For example, inventory tracking by companies using QR codes and barcodes can be replaced with unique

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

formations of points, whereby the features of polygons represent important information. If integrated with a mobile based intra-organisation system, a significant cut-down on scanner costs may be achieved. The above mentioned method can also be used for star constellation recognition, plant leaf recognition, ear identification and other biometric fields as well.

REFERENCES

1. A.K Jain, Salil Prabhakar, Lin Hong, Sharath Pankanti, "Filter-bank-based fingerprint matching", IEEE transactions on Image Processing, vol. 9, no. 5,pp. 846-859, May 2000.
2. N.K Ratha, V.D Pundit, R.M Bolle, V. Vaish, "Robust Fingerprint Authentication Using Local Structural Similarity," Workshop on Applications of Computer Vision, pp: 29-34, 2000.
3. Z.M. Kovacs-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping", IEEE Transactions on Pattern Anal. Mach. Intel., vol. 22 , no. 11, pp. 1266-1276, 2000.
4. A.A. Saleh, R.R. Adhami, "Curvature-based matching approach for automatic fingerprint identification", Proceedings of the Southeastern Symposium on System Theory, pp. 171-175, 2001.
5. X. Jiang, W.Y. Yau, "Fingerprint minutiae matching based on the local and global structures", Proceedings of the International Conference on Pattern Recognition, pp. 1038-1041, 2000.
6. Xuejun Tan, Bir Bhanu, "Fingerprint matching by genetic algorithms", Pattern Recognition (Elsevier) 39, pp. 465 – 477, 2006.
7. Mridula, Priyanka, "A review on classification of fingerprint images", IOSR Journal of Electronics and Communication Engineering, vol.9, issue 3, pp. 61-66, May-June, 2014.
8. Sunny Arief Sudiro, Rudi Trisno Yuwono, "Adaptable fingerprint minutiae extraction algorithm based on crossing number method for hardware implementation using FPGA device", vol. 2, no. 3, June, 2012.
9. J. Zhou, F. Chen, J. Gu, "A novel algorithm for detecting singular points from fingerprint images", IEEE Trans Pattern Anal Mach Int, vol. 31, no.7, pp. 1239–1250, July 2009.