Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

A Fingerprint Matching Scheme Based on Gradient Difference Compatibility

Preeyakorn Tipwai, Panudech Thipaksorn, and Suthep Madarasmi
King Mongkut's University of Technology Thonburi
91 Prachautid Rd., Bangmod, Toong-kru, Bangkok 10140, THAILAND
neng@cpe.kmutt.ac.th, panudech@rmutl.ac.th, suthep@kmutt.ac.th
http://www.cpe.kmutt.ac.th

Abstract

This paper proposes a new minutiae-based fingerprint matching algorithm using a variation of the Generalized Hough Transform called MGHT, which allows variations in translation, rotation, scale, and some distortions of the fingerprints, with very low complexity. A simple local structure of the fingerprint is constructed to collect template data from a number of minutiae. The matching process is performed by voting for transformation parameters by using target fingerprint local structure to look up similar data in the template. The result with a strong peak in the parameter space is considered as a match. The algorithm is efficient, precise, and robust to noise. The test results on several images demonstrate the effectiveness of this fingerprint matching method.

1. Introduction

AMONG all the biometric techniques, fingerprint-based identification is the oldest method which has been successfully used in numerous applications. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows as well as the minutiae points. Minutiae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending. Fingerprint matching techniques can be placed into two categories: correlation-based and minutiae-based. Correlation-based techniques require the precise location of a registration point and are affected by image translation and rotation. Minutiae-based techniques first find minutiae points and then map their relative placement on the finger. Correlation-based ones are less interesting due to low accuracy; while most applications use minutiae-based to achieve effective fingerprint matching.

In most minutiae-based finger print matching, it is necessary to register two fingerprints to bring the features from the template in spatial proximity of their corresponding counterparts from the query fingerprint. Once the fingerprints are aligned properly with the minutiae features, a global matching can be performed.

Zhang and Wang [1] introduced the use of a core-based structure matching algorithm. First, they used a core detection algorithm to get the core position. Then they define some local structure of the core area. Using these local structures, they can find some corresponding points of the two fingerprint images. The corresponding points in the first stage are then used to match the global features of the fingerprint. Lee *et al.* [2] presented a minutiae-based fingerprint matching algorithm using local alignment. In matching one reference minutiae pair, the reliability of a minutia decreases as the distance from the minutia to the minutia used for alignment increases.

Chen *et al.* [3] presented the use of normalized fuzzy similarity algorithm after matching local topological structures which are constructed using neighboring minutiae surrounding each reference minutia by considering the distance and angles between them. Feng *et al.* [4] introduced a concept of compatibility to the minutiae triangle structure by matching neighboring structure pairs in the query and the template images. Then, a relaxation process is adopted to adjust the similarity matrix of the minutiae triangle cell. Wang *et al.* [5] proposed topology-based algorithms which use the Delaunay triangle edge as the comparison index to obtain the transformations $(\Delta\theta, \Delta x, \Delta y)$ by a local matching of control points in the minutiae set. The transformations are used to apply the Radial basis functions (RBF) for non-rigid deformations. Then, a bipartite matching scheme is applied to improve the matching accuracy.

Gu *et al.* [6] present a fingerprint representation including both the global structure of the orientation fields and the minutiae as local cues. The generalized Hough transform (GHT) [7] is used to find a number of candidates for geometrical transformation parameters (t_x, t_y, θ) which specify the translation and rotation. The best one is selected to map the template image on the query image to perform a global matching on orientation field.

The classical Generalized Hough Transform (GHT) stores 1) the distance r from reference position of the object to the corresponding pixel on the contour and 2) the angle α between the x-axis and a line drawn from that pixel to the reference point in a table called the 'R-table'.

The R-table is indexed by each computed gradient value. In the detection process the test image with gradient values for the entire image is used to look up elements in the R-table to perform the voting in a 2-dimension Hough space, $A(x_c, y_c)$. GHT can be modified to allow transformations in scale S and rotation β at the expense of increasing to a 4-dimensional accumulator array $A(x_c, y_c, S, \beta)$. The resulting algorithm has a complexity of $O(N^2RS)$ with poor search results as it is impractical to perform peak clustering in 4 dimensions.

Due to differences in position and pressure while making each fingerprint imprint, fingerprint images often suffer from translation, rotation, and scaling transformation. Most previous local alignment algorithms are computationally expensive due to the complexity in local structures. Moreover, most algorithms do not consider the scaling transformation, which, in fact, is often necessary; for instance, in the case a different scanner of varying resolution is used.

In this paper, we propose an efficient fingerprint-matching algorithm using a modified generalized Hough transform (MGHT) which was first presented by one of the authors of this paper [8][9]. The MGHT allows variations in translation, rotation, scale, and some distortions of the fingerprints. It uses a 2-dimensional accumulator array to automatically search non-parametric objects in a target image that allows for different size and orientation relative to the trained, target object. The difference in gradient direction of each edge pair is invariant to rotation of the object. Therefore, the normal direction at each contour point and at its opposite contour point, are kept as the information including the distance between the 2 ends. The 2-dimensional Hough parameter space reduces the complexity of the GHT algorithm from $O(N^2RS)$ to $O(N^2)$, while remaining rotation and scale invariant.

2. Fingerprint Preprocessing

Prior to matching, the input gray-scale fingerprint is normalized to remove the effects of sensor noise and finger pressure differences for each fingerprint. Then, thinning converts the gray-scale image to a binary thinned-contour image with a single pixel width as shown in figure 1(b). The tangent direction T, which has the value between -90° and 90°, is calculated for each pixel. During both the learning and matching processes, the normal directions θ can be obtained as $T \pm 90°$.

3. Fingerprint Template Feature Extraction

Given a thinned-contour fingerprint template with the tangent field T, we take the image center (X_C, Y_C) as the reference point. Then, the feature set of the template is stored in an R-Table, which is formed by collecting the difference of the normal directions, $\Delta\theta$, for those edge pairs which face each other in each normal direction, $\theta = T \pm 90$. From one starting edge point, a line is traversed in direction of θ_1 degrees to hit several contour points until it meets the image boundary. For each intersected edge point, a feature vector is extracted as illustrated in figure 2. As illustrated in figure 3, the feature vector is kept in the R-Table as a relation $\Delta\theta \rightarrow (r, \alpha, \theta, L)$, where $\Delta\theta = \theta_2 - \theta_1$ with θ_2 as the normal at the intersected point, L is the distance between the endpoints, r is the distance between the starting contour point and the arbitrary reference point (X_C, Y_C), and α is the angle from the starting contour point to (X_C, Y_C) with respect to the image x-axis. Thus, an element in the R-Table contains the information of one pixel and an intersected edge for later use to compute the scale and rotation values in the detection process as presented in the next section.

(a) (b) (c)

Figure 1. Fingerprint Preprocessing. (a) Original Fingerprint. (b) Thinned contour Fingerprint (c) Tangent T and normal angle θ

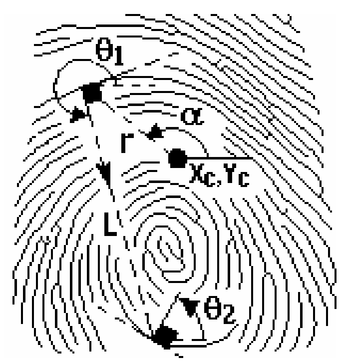


Figure 2. Template Fingerprint A minutia feature vector constructed by p_1 and one of its opposite point p_2, and the reference point of the fingerprint (X_C, Y_C). A line is extended from p in the γ direction with respect to the contour normal θ_1 to meet several edges, to get a second normal angle θ_2.

$\Delta\theta$	$r_1, \alpha_1, \theta_1, L_1$	$r_2, \alpha_2, \theta_2, L_2$	$r_3, \alpha_3, \theta_3, L_3$
0…19	15,180,195,99	15,179,219,101	16,177,216,102
20…39	17,160,23,5	14,159,38,7	18,161,175,62
30…49	19,165,31,53	20,170,8,52	22,167,15,52
…	…	…	…
340…359	23,105,346,11	24,103,165,11	21,102,346,18

Figure 3. Illustration of the R-Table

The algorithm for creating the R-Table is as follows:

1) Start with a thinned fingerprint image and compute the tangent direction T for each 'on', edge pixel
2) Calculate the normal directions $\theta = T \pm 90$ degrees
3) Pick an arbitrary reference point (X_c, Y_c), such as the center of the image.
4) For each 'on' pixel (x, y) in each normal direction θ extend a line in the θ direction until it hits another contour (if any). For each intersected edge (x_2, y_2):
   4.1 Save the length L of this line, the normal θ_1 of start point, and the normal θ_2 of the end point.
   4.2 Draw a line from (x, y) to the reference point (X_c, Y_c). Note the length r of this line and its angle α with the x-axis.
   4.3 Enter the value in the r-table $\Delta\theta \rightarrow (\theta, L, \alpha, r)$ where $\Delta\theta = \theta_1 - \theta_2$ and θ is θ_1

4. Fingerprint Matching

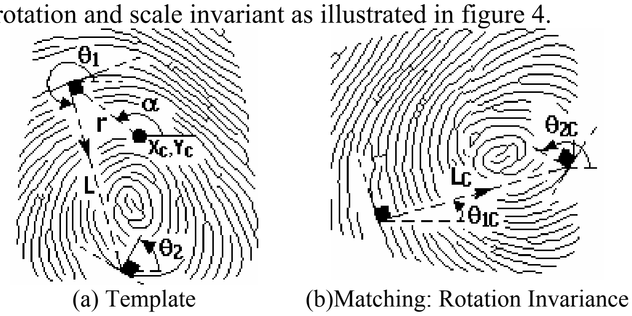
In the detection process, for each edge pixel (x, y), a line is traversed in the θ_{1C} degree to hit several points until reaching the image boundary. At each hit edge point, the normal value θ_{2C} and the distance L_C between the two edge endpoints is computed. Then, the difference in normal directions $\Delta\theta_C = \theta_{1C} - \theta_{2C}$ is used to look up the feature vectors values $\Delta\theta_C \rightarrow (\theta, L, \alpha, r)$ from the R-Table. For each element $(\theta_i, L_i, \alpha_i, r_i)$, where i = 1, 2,..N and N is the number of vectors in the indexed row $\Delta\theta_C$, the scale size S and rotation angle β are calculated using equations (1) and (2), respectively. Also, a new reference point (X_C, Y_C) is computed using equation (3). All the calculated values are used to update the parameter Hough space $A(X_C, Y_C)$.

$$S_i = L_C / L_i \tag{1}$$

$$\beta_i = \theta_{1C} - \theta_i \tag{2}$$

$$X_C = x + rS_i \cos(\alpha_i + \beta_i), Y_C = y + rS_i \sin(\alpha_i + \beta_i) \tag{3}$$

This MGHT reduces the complexity of the GHT algorithm from O(N^2RS) to O(N^2), while remaining rotation and scale invariant as illustrated in figure 4.



(a) Template
Δθ=60°-290°=-230°=130°

(b)Matching: Rotation Invariance
Δθ = 150° - 20° = 130°

Figure 4 Compatible Gradient Difference. In spite of scale and rotation changes, the gradient difference of the contours at opposite ends remains the same.

The matching algorithm is:

1) Given a thinned fingerprint query image with computed tangent direction T for each edge pixel
2) Calculate the normal directions $\theta = T \pm 90$ degrees
3) For each 'on' edge pixel in a normal direction θ Extend a line in the θ direction until it hits another contour (if any). For each intersected edge (x_2, y_2):
   3.1 Save the length L of this line, the normal of start point as θ_{1C}, and that of end point as θ_{2C}. Let $\Delta\theta_C = \theta_{1C} - \theta_{2C}$.
   3.2 For each value in R-table where $\Delta\theta \approx \Delta\theta_C$, get the corresponding values $\Delta\theta \rightarrow (\theta_i, L_i, \alpha_i, r_i)$ where i = 1, 2,..N for N such values. For each i, find (X_c, Y_c) using Equations (1), (2), and (3) then increase the accumulator array A(X_c, Y_c).
4) Find the local maximum in the Hough parameter space where a high peak in A(X_c, Y_c) indicates a potential fingerprint match.

5. EXPERIMENTAL RESULTS

Two classes of experiments were conducted to test our algorithm. The fingerprint image template and target image used in the first class of experiments were clear images. The second class of experiments used images for matching from our fingerprint scanners.

5.1. Experiments using Clear Fingerprint Images

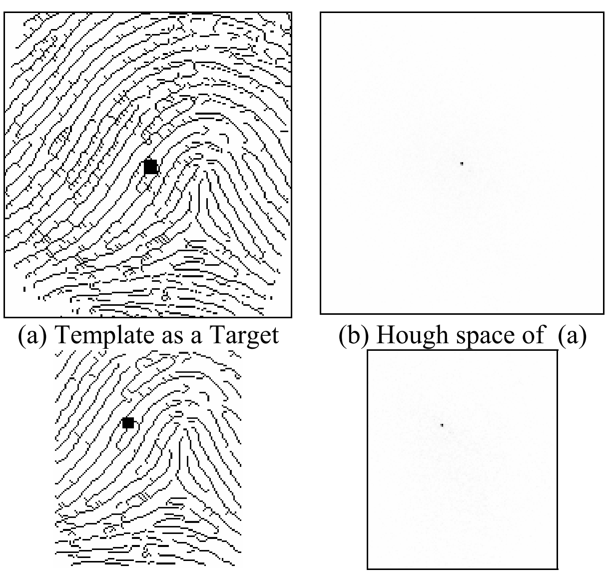
To reduce the number of edges used, only 20% of the target edges are used to vote in the Hough space. The target database, shown in figure 5, consists of 30 images from 17 different fingerprints. Some images are cropped from the original image to test our algorithm under various input conditions.

In figure 6, we show 2 examples of a correct match, one when the target is exactly the same image as the template and another when the target is a cropped version of the template. By considering the Hough space we are able to determine whether the target images have same feature measures as the template. If there is a single bright spot in the Hough space as shown in figure 6 (b) and (d), the fingerprint image is considered identical to the template image. Although they are cropped from the template image, the single bright spot in the figure 6(d) is still reported, indicating a correct match. In figure 7, we show 3 examples of a detected match, when the targets are different transformed versions of the template.

We then display examples of cases with no match in figure 8. In figure 8 (b) and (d), the vote in the Hough space is spread out with no distinct peak. It means the query fingerprint image is very different from the template.

Figure 9 shows votes of fingerprint matching in the database with 3 correct targets. The result correctly shows a peak in 3 images which are scaled, cropped, and rotated versions of the template.

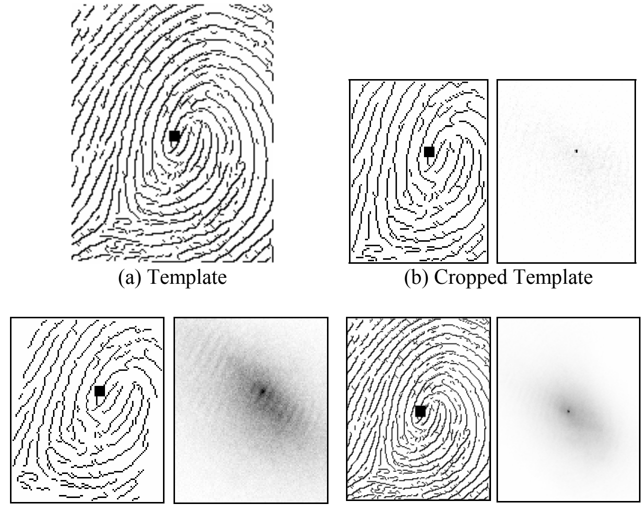
Figure 5. 30 samples from 17 different fingerprint samples.



(a) Template as a Target (b) Hough space of (a)

(c) Target from cropped template (d) Hough space of (c)

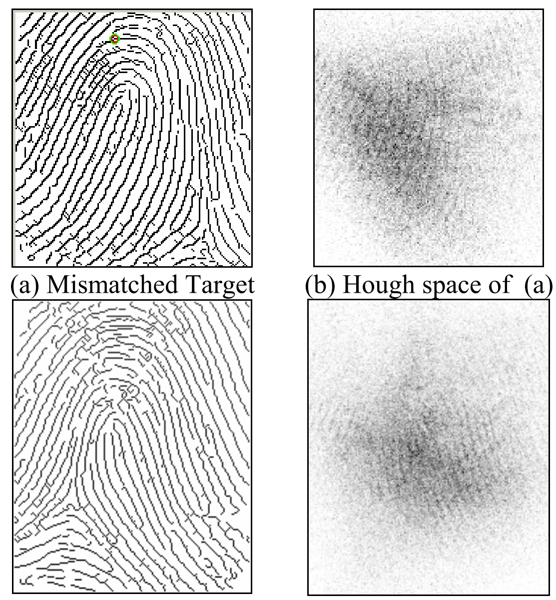
Figure 6. Examples of a detected match.



(a) Template (b) Cropped Template

(c) Cropped and Rotated Template (d) Scaled Template

Figure 7. Examples of detected matches for different images.



(a) Mismatched Target (b) Hough space of (a)

(c) Mismatched Target (d) Hough space of (c)

Figure 8. Examples of cases when target does not match.

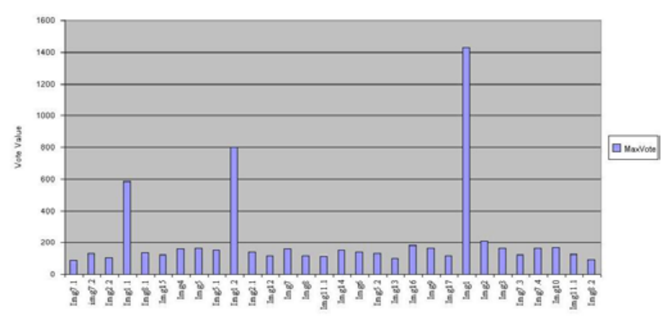


Figure 9. Max vote graph for first experiment's dataset.

5.2. Experiments using Scanned Fingerprints

In this experiment, the fingerprint image template and the target image were collected from two fingerprint scanners M1 and M2. They consist of 40 images from 5 people. Some fingerprint samples are shown in figure 10. Table 1 shows the type of experiments conducted to test which edge sampling method is optimal.

Figure 10. Examples of fingerprint images from fingerprint scanners.

The selected template are shown in figure 11. Note that each matched target does show as a maximum peak in the Hough Space. The results in figure 12 are obtained from two machines. On the left hand side is the result of fingerprint machine M1 and on the right hand side is the result of M2. The Hough space shows that some vote points are scattering, although there is a distinct peak at the correct point. The correct position is detected, although the 2 fingerprints are from different machines.

TABLE 1. FOUR SETS OF EXPERIMENTS ON SCANNED FINGERPRINTS

Experiment	Template Sampling	Target Sampling
B1	30 degrees per step	20% of fingerprint
B2	30 degrees per step	10 degree per step
B3	45 degrees per step	20% of fingerprint
B4	45 degrees per step	10 degree per step

Figure 13 shows the scattered Hough space. There are no clear peaks, meaning that a match with the target image is not found.

In this experiment, there are peaks showing the correct match to each experiment as shown in figure 14.

In figure 15, the processing time of each target image is shown to compare this aspect. The computational time depends on the area of the targets. Smaller areas consume lesser time than larger areas.

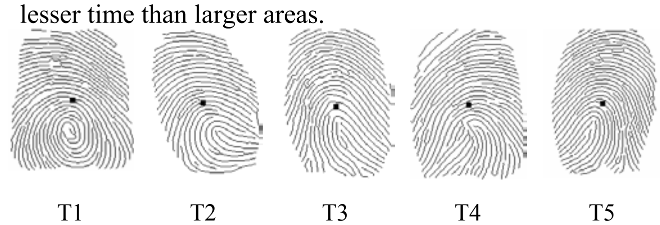


Figure 11. Five examples of fingerprint templates with reference point.

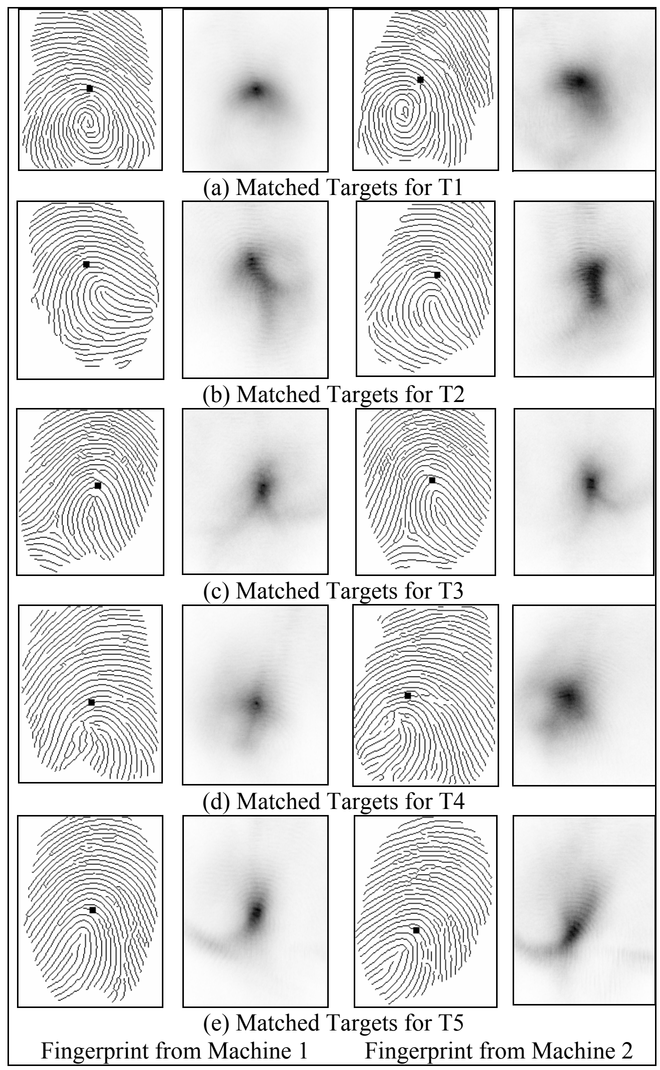


Figure 12. Examples of matched targets for templates T1 – T5 shown in figure 9, with detected reference point.

(a) (b) (c) (d)

Figure 13. Examples when the fingerprints do not match, for template T1. (a) and (c) Mismatched Targets, (b) and (d) Hough space of (a),(c), respectfully.

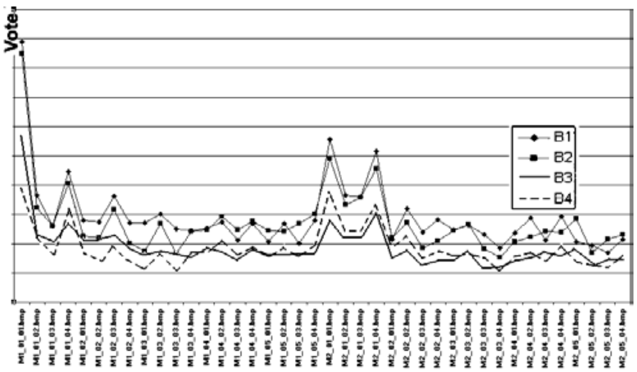


Figure 14. Maximum vote of each experiment B1 to B4.

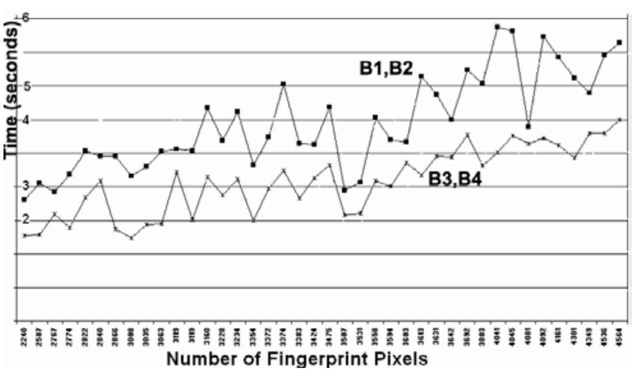


Figure 15. Processing time for each experiment B1 to B4.

5.3. Summary Fingerprint Matching images from Two Machines

Table 2 shows the efficiency of the fingerprint matching. According to the table, the parameter setting for B4 is most accurate since this setting collects data from the original images in every 45 degrees, i.e. eight ways around the reference (Xc, Yc). It is enough for the matching process. In addition, collecting data from the target images by changing the tangent value in every ten degrees is more accurate than sampling data by 20 percent of the images

6. Conclusion

We have applied the Modified Generalized Hough Transform (MGHT) on fingerprint matching. Shape and size of lines in fingerprint are used for learning and comparing in our method. The algorithm was tested for several cases.

According to the results, the method has a correctness of 70% to 100% and the most suitable parameter for the matching is to collect data in eight directions for learning. To match the original image with the target images, the data should be collected when the tangent value is changed by 10 degrees.

TABLE 2 EFFICEINCY OF ALL EXPERIMENTS

	B4	B3	B2	B1
Correct Match between template and target	28 of 40 70%	22 of 40 55%	16 of 40 40%	23 of 40 57.5%
No. of targets/template	40	40	40	40
Total Incorrect	37 of 200 18.5%	55 of 200 27.5%	34 of 200 17%	26 of 200 13%
Number of Tests (5 templates, 40 targets)	200	200	200	200

References

[1] W. Zhang and Y. Wang, "Core-Based Structure Matching Algorithm of Fingerprint Verification," *IEEE Conf. on Pattern Recognition*, 2002, pp. 70-74.

[2] D. Lee, K. Choi, and J. Kim, "A Robust Fingerprint Matching Using Local Alignment," *IEEE Conf. on Pattern Recognition*, 2002, pp. 803-806.

[3] X. Chen, J. Tian, and X. Yang, "A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure," *IEEE Trans. Image Processing*, Vol. 15, No. 3, 2006, pp. 767-776.

[4] Y. Feng, J. Feng, X. Chen, and Z. Song, "A Novel Fingerprint Scheme Based on Local Structure Compatibility," *IEEE Conf. on Pattern Recognition*, 2006.

[5] C. Wang, M. Gavrilova, Y. Luo, and J. Rokne, "An Efficient Algorithm for Fingerprint Matching," *IEEE Conf. on Pattern Recognition*, 2006.

[6] J. Gu, J. Zhou, and C. Yang, "Fingerprint Recognition by Combining Glocal Structure and Local Cues," *IEEE Trans. Image Processing*, Vol. 15, No. 7, 2006, pp. 1952-1964.

[7] D.H. Ballard: Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, vol. 13, no. 2 ,1981, pp. 111-122.

[8] Nimkerdphol, K. and Madarasmi, S., "A Modified Generalized Hough Transform Algorithm for Image Retrieval by Contour Matching," presented at *ICICS*, Singapore, 2001.

[9] Tipwai, P. and Madarasmi, S., "Image Search Using Deformable Contours," in *Proc ICIP*, 2002, pp. I 437 - I 440.A. Alpher, , and J. P. N. Fotheringham-Smythe. Frobnication revisited. Journal of Foo, 13(1):234–778, 2003.