A Near-Linear Time Algorithm for Binarization of Fingerprint Images using Distance Transform

Xuefeng Liang, Arijit Bishnu, and Tetsuo Asano

JAIST, 1-1, Asahidai, Tatsunokuchi, 9231292, Japan.
{xliang, arijit, t-asano}@jaist.ac.jp

Abstract. Automatic Fingerprint Identification Systems (AFIS) have various applications to biometric authentication, forensic decision, and many other areas. Fingerprints are useful for biometric purposes because of their well known properties of distinctiveness and persistence over time. Fingerprint images are characterized by alternating spatial distribution of gray-level intensity values of ridges and ravines/valleys of almost equal width. Most of the fingerprint matching techniques require extraction of minutiae that are the terminations and bifurcations of the ridge lines in a fingerprint image. Crucial to this step, is either detecting ridges from the gray-level image or binarizing the image and then extracting the minutiae. In this work, we focus on binarization of fingerprint images using linear time euclidean distance transform algorithms. We exploit the property of almost equal widths of ridges and valleys for binarization. Computing the width of arbitrary shapes is a non-trivial task. So, we estimate width using distance transform and provide an $O(N^2 \log M)$ time algorithm for binarization where M is the number of gray-level intensity values in the image and the image dimension is $N \times N$. With M for all purposes being a constant, the algorithm runs in near-linear time in the number of pixels in the image.

1 Introduction

Automatic fingerprint identification systems (AFIS) provide widely used biometric techniques for personal identification. Fingerprints have the properties of distinctiveness or individuality, and the fingerprints of a particular person remains almost the same (persistence) over time. These properties make fingerprints suitable for biometric uses. AFISs are usually based on minutiae matching [9, 14, 17, 18]. Minutiae, or Galton's characteristics [11] are local discontinuities in terms of terminations and bifurcations of the ridge flow patterns that constitute a fingerprint. These two types of minutiae have been considered by Federal Bureau of Investigation for identification purposes [29]. A detailed discussion on all the aspects of personal identification using fingerprint as an important biometric technique can be found in Jain et al. [17, 19]. AFIS based on minutiae matching involves different stages (see Figure 1 for an illustration):
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.

The preprocessing phase is known to consume almost 90-95% of the total time of fingerprint identification and verification [3]. That is the reason a considerable amount of research has been focussed on this area.

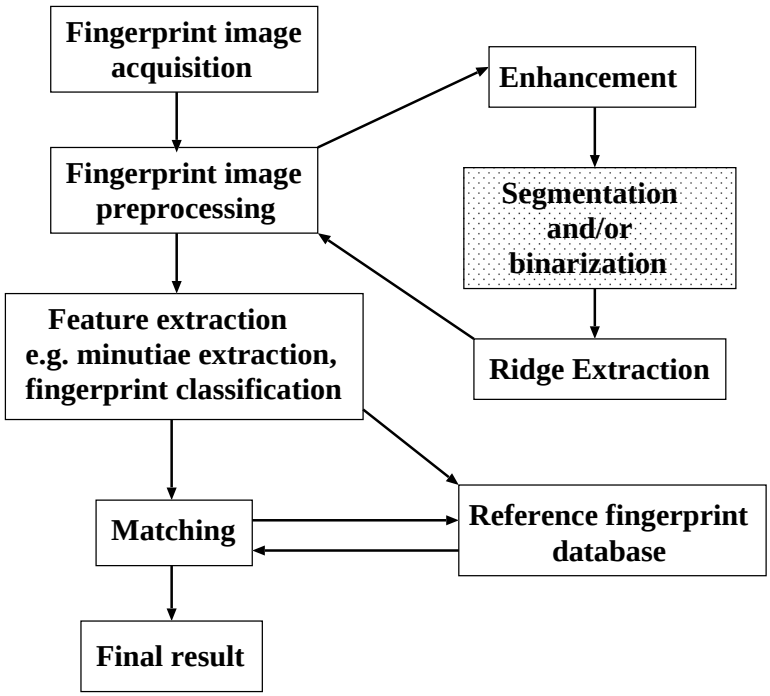


Fig. 1. A flowchart showing different phases of fingerprint analysis. The highlighted module shows the area of our work.

Our work proposed in this paper involves binarization of fingerprint images that is to be preceded by an enhancement step. So, below we discuss briefly enhancement. Also, we briefly discuss and review segmentation and binarization methods applied to fingerprint images.

1.1 Enhancement of Fingerprint Images

Fingerprint images require specialised enhancement techniques owing to their inherent characteristics like high noise content, particular structural content of alternating ridges and valleys. Conventional image processing enhancement techniques are not very suitable for a fingerprint image [8]. Fingerprint image enhancement algorithms are available both for binary and gray level images. A binary fingerprint image consists of ridges marked as object (1) pixels and the

rest as background pixels (0). Hung [8] designed an algorithm for enhancing a binary fingerprint image based on the structural information of its ridges. Ridge widths are normalized based on some region index. Ridge breaks are corrected using the dual relationship between ridge breaks and valley bridges. However, obtaining a binary fingerprint image from a gray-tone image involves inherent problems of binarization and thinning or ridge extraction procedures [6]. Thus, most of the enhancement algorithms are designed for gray-level fingerprint images. The much widely used PCASYS package [5] uses an enhancement algorithm described earlier [1]. It involves cutting out subregions of the images (a 32×32 block to be specific), taking their FFT and suppression of a band of low and high frequency components followed by some non-linear operations in the frequency domain and transforming it back to the spatial domain. This algorithm was also used by Kovács-Vajna [18]. We have also used this enhancement algorithm in our work owing to its simplicity and elegance.

1.2 Segmentation of Fingerprint Images

In literature concerning fingerprints, some authors have used the term segmentation to mean the process of generating a binary image from a gray-level fingerprint image. But, as suggested in [19], the most widely held view about segmentation of a fingerprint image is the process of separation of fingerprint area (ridge, valley and slope areas in between ridge and valley areas) from the image background. The process of segmentation is useful to extract out meaningful areas from the fingerprint, so that features of the fingerprint are extracted from these areas only. Fingerprint images are characterized by alternating spatial distribution of varying gray-level intensity values of ridges and ravines/valley. This pattern is unique to a fingerprint area compared to the background which does not have this spatial distribution of gray-level values. Also, global thresholding for segmentation does not work as the spatial distribution of gray-level values keeping their alternating structure intact, can vary in the absolute magnitude of their gray-level values. Thus, local thresholding is needed. Exploitation of these property have been the key of most of the segmentation algorithms. O'Gorman and Nickerson [24] used a $k \times k$ spatial filter mask with an appropriate orientation based on user inputs for labeling the pixels as foreground (crest) or background. Mehtre and Chatterjee [21] described a method of segmenting a fingerprint image into ridge zones and background based on some statistics of local orientations of ridges of the original image. A gray-scale variance method is used in the image blocks having uniform gray-level, where the directional method of segmentation fails. Ratha et al. [25] used the fact that noisy regions show no directional dependence, whereas, fingerprint regions exhibit a high variance of their orientation values across the ridge and a low variance along the ridge to design a segmentation algorithm that works on 16×16 block. Maio and Maltoni [20] used the average magnitude of gradient values to discriminate foreground and background regions. The idea behind this is that fingerprint regions are supposed to have more edges than background region and as such would have higher gradient values.

1.3 Binarization of Fingerprint Images

The general problem of image binarization is to obtain a threshold value so that all pixels above or equal to the threshold value are set to object pixel (1) and below the threshold value are set to background (0). Thresholding can be done globally where a single threshold is applied globally or locally where different thresholds are applied to different image regions. Images, in general, have different contrast and intensity, and as such local thresholds work better. The thresholding problem can be viewed as follows. Given an image I with $N \times N$ pixel entries, and gray-level intensity value g ranging from 0, 1, ... to $M-1$, select a value $t \in [0, M-1]$ based on some condition so that a pixel (i, j) is assigned a value of 1 if the gray-level intensity value is greater or equal to t, else assign 0 to the pixel (i, j). The condition mentioned above is decided based on the application at hand. The binarization methods applicable to fingerprint images draw heavily on the special characteristics of a fingerprint image. Moayer and Fu [23] proposed an iterative algorithm using repeated convolution by a Laplacian operator and a pair of dynamic thresholds that are progressively moved towards an unique value. The pair of dynamic thresholds change with each iteration and control the convergence rate to the binary pattern. Xiao and Raafat [30] improved the above method by using a local threshold, to take care of regions with different contrast, and applied after the convolution step. Both of these methods requiring repeated convolution operations are time consuming and the final result depends on the choice of the pair of dynamic thresholds and some other design parameters. Coetzee and Botha [7] proposed an algorithm based on the use of edges in conjunction with the gray-scale image. The resultant binary image is a logical OR of two binary images. One binary image is obtained by a local threshold on the gray scale image and the other binary image is obtained by filling in the area delimited by the edges. The efficiency of this algorithm depends heavily on the efficiency of the edge finding algorithm to find delimiting edges. Ratha et al. [25] proposed a binarization approach based on the peak detection in the gray-level profiles along sections orthogonal to the ridge orientation. The gray-level profiles are obtained by projection of the pixel intensities onto the central section. This heuristic algorithm though working well in practice has a deficiency that it does not retain the full width of the ridges, and as such is not a true binary reflection of the original fingerprint image.

In this work, we propose a combinatorial algorithm for binarization of fingerprint images based on Euclidean distance transform. Most of the previous algorithms discussed here are heuristics in that they do not start with a definition of an optimal threshold. In contrast, we define a condition for an optimal threshold based on equal widths of ridges and valleys. We show how distance transform can be used as a measure for width and then design an algorithm to efficiently compute the threshold for binarization. Using distance transform for binarization has also got another distinct advantage. The next step following binarization is ridge extraction and ridges can be efficiently extracted using distance transform values. As the same feature can be used for both binarization and ridge extraction, a lot of time savings can be obtained in real applications.

The rest of the paper is organised as follows. In Section 2, we briefly review Euclidean Distance Transform algorithm. Section 3 has a discussion on measuring width of shapes using average Distance Transform values. Section 4 discusses the threshold criteria and discusses the algorithm for thresholding and shows results on different fingerprint images. Finally, we finish with some discussions in Section 5.

2 Distance Transform

A two-dimensional binary image I of $N \times N$ pixels is a matrix of size $N \times N$ whose entries are 0 or 1. The pixel in a row i and column j is associated with the Cartesian co-ordinate (i, j). For a given distance function, the *Euclidean distance transform* of a binary image I is defined in [4] as an assignment to each background pixel (i, j) a value equal to the Euclidean distance between (i, j) and the closest feature pixel, i.e. a pixel having a value 1. Breu et al. [4] proposed an optimal $O(N \times N)$ algorithm for computing the *Euclidean distance transform* as defined using Voronoi diagrams. Construction and querying the Voronoi diagrams for each pixel (i, j) take time $\theta(N^2 \log N)$. But, the authors use the fact that both the sites and query points of the Voronoi diagrams are subsets of a two-dimensional pixel array to bring down the complexity to $\theta(N^2)$. In [13], Hirata and Katoh define *Euclidean distance transform* in an almost same way as the assigment to each 1 pixel a value equal to the Euclidean distance to the closest 0 pixel. The authors use a bi-directional scan along rows and columns of the matrix to find out the closest 0. Then, they use an envelope of parabolas whose parameters are obtained from the values of the bi-directional scan. They use the fact that two such parabolas can intersect in at most one point to show that each parabola can occur in the lower envelope at most once to compute the *Euclidean distance transform* in optimal $\theta(N^2)$ time. In keeping with the above, we define two types of *Euclidean distance transform* values. The first one $DT_{1,0}$ is the same as the above. The second one is $DT_{0,1}$ which is the value assigned to a 0 pixel equal to the Euclidean distance to the nearest 1 pixel. Using the results given in [13], we have the following fact:

Fact 1 *Both $DT_{1,0}$ and $DT_{0,1}$ can be computed in optimal time $O(N^2)$ for an $N \times N$ binary image. Also, the values of both $DT_{1,0}$ and $DT_{0,1}$ are greater than or equal to* 1.

3 Distance Transform and Width

The fingerprint images are characterized by almost equal width ridges and valleys as shown in Figure 2. We will use this particular characteristic of the fingerprint image for binarization. Measuring the width for arbitrary shapes is a difficult, non-trivial problem. In this section, we model the problem in a continuous domain to show how distance transform can be used to find equal width ridges and valleys.

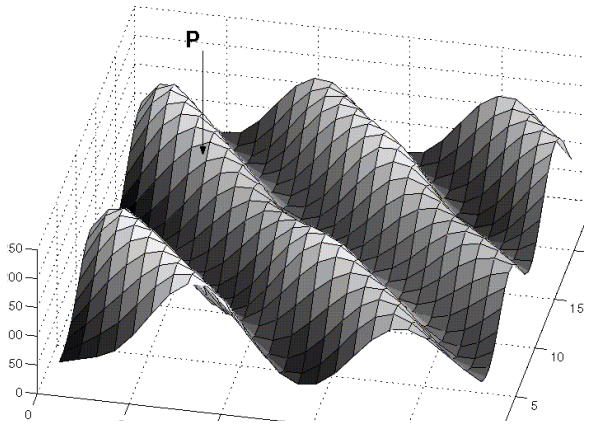


Fig. 2. Magnified view of a part of the gray scale topology of a fingerprint image.

3.1 Model in the continuous domain

The fingerprint image can be modeled as shown in Figure 3. In the continuous domain, the image is a continuous function $f : (x, y) \rightarrow I\!R$. A cross section of this function along a direction perpendicular to the ridge increases till it reaches the ridge point which is a maxima, then decreases till it reaches the valley, which is a minima; and this cycle repeats. Let $t \in [0, M]$ be a threshold, such that if f is thresholded at t, and if the value of f is greater than t, it is mapped to 1, else to 0. See Figure 3. The highlighted part shown on the right is the part mapped to 1. After thresholding, the parts would be rectangles as shown in Figure 3. We compute the total distance transform values of the rectangles. Consider a rectangular object $ABCD$ of width w and height h, with $h > w$. The medial axis of this object is given by the line segments $\overline{AE}$, $\overline{BE}$, $\overline{EF}$, $\overline{FD}$, $\overline{FC}$. The medial axis divides the rectangular shape into four regions such that the nearest boundary line from any point in the region is determined. As an example, the region 1 has $\overline{AD}$ as its nearest boundary line and region 3 has $\overline{AB}$ as its nearest boundary line. The total distance transform value for region 1 is $\int_0^{w_i/2} \int_{-y+w_i/2}^{y+(h-w_i/2)} (w_i/2 - y)\, dxdy = (w_i^2 h)/8 - w_i^3/12$. Similarly, the total distance transform value for region 3 is $\int_{x-w_i/2}^{-x+w_i/2} \int_0^{w_i/2} x\, dxdy = w_i^3/24$. So, the total distance transform value $\phi_{dt}(w_i)$ of the rectangle is $w_i^2 h/4 - w_i^3/12 = w_i^2/4(h - w_i/3)$. Note that, the total distance transform value increases (decreases) with the increase (decrease) of width because $\phi_{dt}(w_i)' > 0$ and $h > w$. Now, the total distance transform $DT_{1,0}$ is $w_1^2 h/4 - w_1^3/12 + w_3^2 h/4 - w_3^3/12$ and the total distance transform $DT_{0,1}$ is $w_2^2 h/4 - w_2^3/12$. Now, as t increases, both

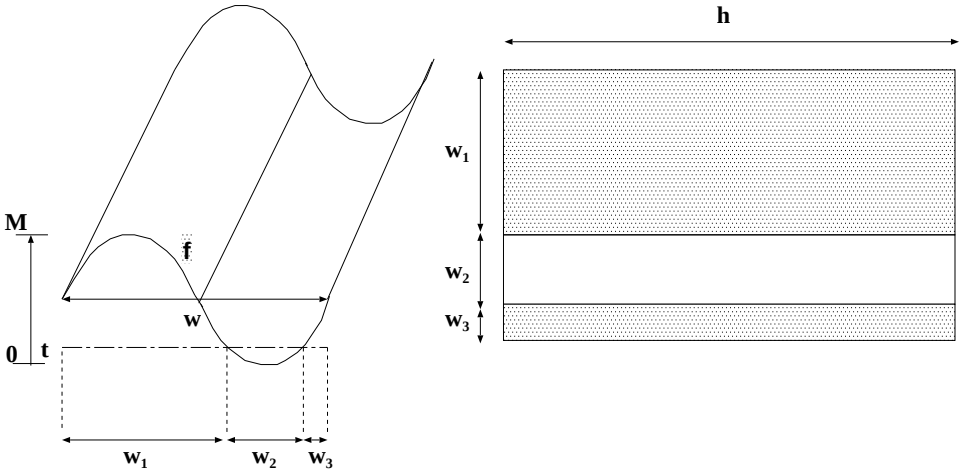


Fig. 3. Diagram of the model.

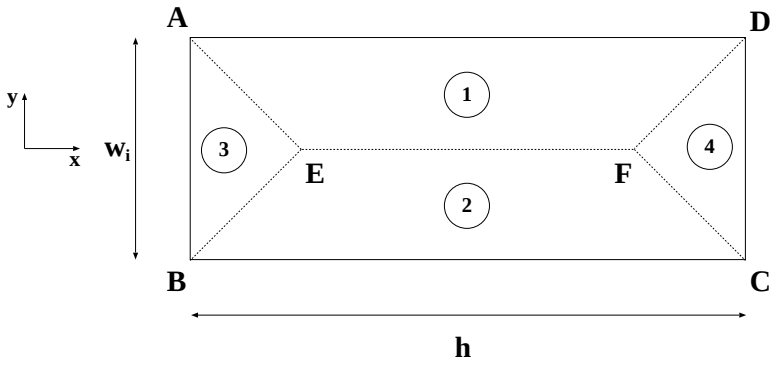


Fig. 4. Diagram for computing total distance transform.

w_1 and w_3 decrease and w_2 increases. This implies that with increase of t, $DT_{1,0}$ decreases and $DT_{0,1}$ increases. So, $DT_{1,0}$ and $DT_{0,1}$ can intersect only once and evidently, $DT_{1,0}$ is equal to $DT_{0,1}$ when $w_1 = w_2$. That is, the optimal value of threshold is reached when $DT_{0,1} = DT_{1,0}$, implying $w_1 = w_2$. This simple analysis shows that total distance transform can be used as a measure of finding a threshold that gives equal width ridges and valleys. Our goal in this work is to find an optimal threshold to binarize the fingerprint image. The optimality criteria is given by the equal width of ridge and valley. So, more formally we have the following definition.

Definition 1. *The optimal threshold is a value* $t \in [0, M]$ *that binarizes the image such that the ridge width is equal to the valley width or sum total of distance transform values are equal.*

3.2 Discrete image and distance transform

In the discrete model, the co-ordinates are discrete given by the pixel locations. The gray-level values g are also discrete taking values from 0 to $M-1$. So, the observations from the previous subsection do not directly apply. But, the crucial observation from the previous subsection is that sum total of $DT_{1,0}$ values decreases with t and the sum total of $DT_{0,1}$ values increases with t. Then, the optimal threshold t can be obtained as that value of t that makes the width of the 1 region and 0 region equal and can be computed from the intersection of the curves of the sum total of $DT_{0,1}$ and $DT_{1,0}$ values. For, the analysis, we make the following assumption. The pixels take the gray-level intensity values such that all the intermediate gray-level values between the maximum and the minimum are present. With that assumption, we have the following lemma.

Lemma 1. *The sum total of $DT_{1,0}$ values decreases with the threshold t. Similarly, the sum total of $DT_{0,1}$ values increase with the threshold t.*

The proof is easy. We know that each of the Euclidean distance transform values in the discrete domain is greater than or equal to 1 (see Fact 1). So, with the threshold t increasing, pixels in the binary image move from the regions of 1 to 0, thus making $DT_{1,0}$ and $DT_{0,1}$ decreasing and increasing respectively. Also, note that the assumption that the pixels take the gray-level intensity values such that all the intermediate gray-level values between the maximum and the minimum are present, ensures the strictly decreasing and increasing relations of sum total of $DT_{1,0}$ and $DT_{0,1}$ values. Otherwise, it would have been non-increasing and non-decreasing respectively.

Also, in the discrete case, we may not be able to locate a single value, where the functions of sum total of $DT_{1,0}$ and $DT_{0,1}$ meet. So, we modify the definition of the optimal threshold in the discrete case as follows.

Definition 2. *The optimal threshold can be two values t_1 and t_2 such that $t_2 - t_1 = 1$ and the sum total of $DT_{1,0}$ values is greater than the sum total of $DT_{1,0}$ values at t_1 and their relation reverses at t_2.*

With this definition in place, we are in a position to design the algorithm in the next section.

4 Algorithm and Results

4.1 Algorithm for Binarization

To take care of different contrast and intensity across different image regions, we apply local thresholding. We cut out sub-blocks of image region and apply the enhancement algorithm due to [1] followed by our binarization algorithm.

Algorithm for binarization

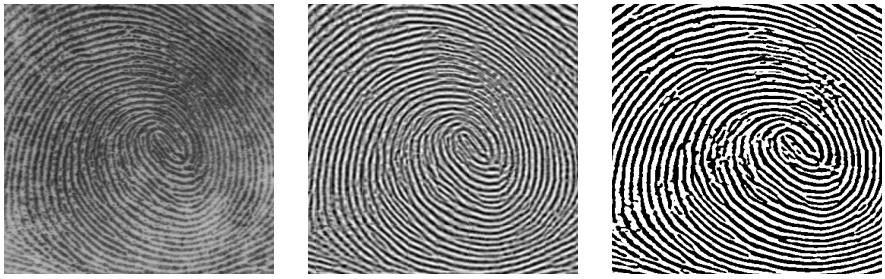
Input: A gray-level fingerprint image I with gray-level intensity varying from 0 to $M-1$, and of size $N \times N$;
Output: A thresholded binary image

1. **do for all** sub-block B_i of the image I;
2. Apply the enhancement algorithm given in [1];
3. $t_1 = 0$, $t_2 = M - 1$; $mid \leftarrow \lceil (t_1 + t_2)/2 \rceil$;
4. **do**
5. $mid \leftarrow \lceil (t_1 + t_2)/2 \rceil$;
6. Compute $SumDT_{1,0}^{mid}$ and $SumDT_{0,1}^{mid}$;
7. **if**$(SumDT_{1,0}^{mid} > SumDT_{0,1}^{mid})$ $t_1 \leftarrow mid$;
8. **else** $t_2 \leftarrow mid$;
   while$(t_2 - t_1 > 1)$
9. Threshold obtained for binarization is t_1 or t_2;

The loop originating in Step 4 runs $O(\log M)$ times and the dominant computation is the computation of Euclidean Distance Transform and its sum which takes $O(N^2)$ time (see Fact 1). Thus the total time complexity of the binarization process is $O(N^2 \log M)$. With M, the number of gray-levels, being a constant for all practical purposes, the algorithm for binarization runs in time that is linear in the number of pixel entries which is $O(N^2)$.

4.2 Results on Fingerprint Images

We used the fingerprint images from (i) NIST Special Database 4[28], (ii) NIST Special Database 14[5], (iii) Database B1 of FVC2000[10], and (iv) Database B2 of FVC2000[10]. The images of (i) and (ii) are of size 480×512. The images of (iii) are of size 300×300 and (iv) are of size 364×256. All of the images are of 500 dpi resolution. Figures 5-8(a) show the original image, Figures 5-8(b) show the enhanced image due to [1] and Figures 5-8(c) show the resultant binary image obtained by application of our algorithm.

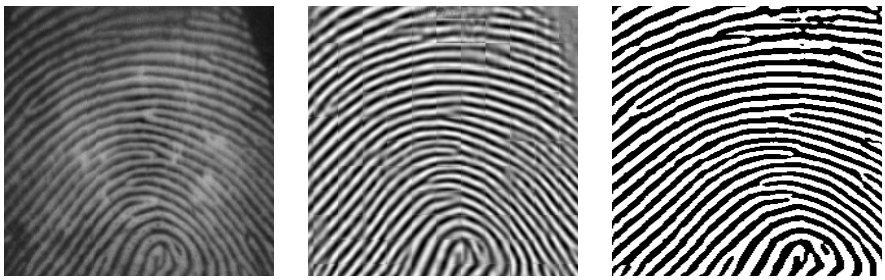


(a) Original image. (b) Enhanced image. (c) Binary image

Fig. 5. Binarization on an image sample from NIST-4 fingerprint image database.

(a) Original image. (b) Enhanced image. (c) Binary image

Fig. 6. Binarization on an image sample from NIST-14 fingerprint image database.

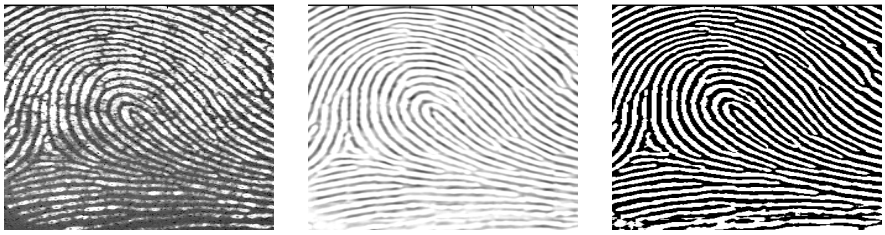


(a) Original image. (b) Enhanced image. (c) Binary image

Fig. 7. Binarization on an image sample from FVC2000 DB1 fingerprint image database.

5 Discussions and Conclusions

We have developed a combinatorial algorithm for binarization of fingerprint images expoiting the fingerprint characteristics of equal width ridge and valleys. We used Euclidean Distance Transform as a measure of width as determining width for arbitrary discrete shapes is a non-trivial task. We have reported relevant results from standard image databases widely used. But, the definition 2 used for our algorithm has a drawback in realistic terms. During the acquisition of fingerprints, ridges, being the elevated structures on the finger, exert more pressure on the device making the acquisition. And as such, the widths of the ridges should be greater than the width of the valley for a more realistic model. But, still the lemma 1 will hold and the algorithm instead of trying to find the crossover point of sum total of $SumDT_{1,0}$ and $SumDT_{0,1}$ will terminate when $SumDT_{1,0}$ is greater than $SumDT_{0,1}$ by a certain ϵ. Determining this ϵ from real fingerprint images is a future problem we would like to address. Also, note that our binarization algorithm using distance transform has a distinct benefit. Please refer to Figure 1. The module following binarization is ridge extraction. Ridge is the skeleton of the thick binary structures obtained from the binarization. Euclidean Distance Transform can be effectively used to find the skeleton [31].



(a) Original image. (b) Enhanced image. (c) Binary image

Fig. 8. Binarization on an image sample from FVC2000 DB2 fingerprint image database.

Thus the same feature of distance transform can be used for both binarization and ridge extraction which in real applications can save a lot of time.

6 Acknowledgment

This research for the first author was conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology. This research for the third author was partially supported by the same Ministry, Grant-in-Aid for Scientific Research (B) and Exploratory Research.

References

1. *Automated Classification System Reader Project (ACS)*, Technical Report, DeLaRue Printrak Inc., Feb., 1985.
2. Bhanu, B. and Tan, X., "Fingerprint Indexing Based on Novel Features of Minutiae Triplets", *IEEE Trans. PAMI*, vol. 25, no. 5, pp. 616-622, 2003.
3. Blue, J. L., Candela G. T., Grother, P. J., Chellappa, R., Wilson, C. L., and Blue, J.D., "Evaluation of Pattern Classifiers for Fingerprint and OCR Application", *Pattern Recognition*, vol. 27, no. 4, pp. 485-501, 1994.
4. Breu, H., Gil, J., Kirkpatrick, D., and Werman, M., "Linear Time Euclidean Distance Transform Algorithms", *IEEE Trans. PAMI*, vol. 17, no. 5, pp. 529-533, 1995.
5. 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.
6. Chang, J. -H., Fan, K. -C., "Fingerprint Ridge Allocation in Direct Gray-Scale Domain", *Pattern Recognition*, vol. 34, no. 10, pp. 1907-1925, 2001.
7. Coetzee, L., and Botha, E. C., "Fingerprint Recognition in Low Quality Images", *Pattern Recognition*, vol. 26, no. 10, pp. 1441-1460, 1993.
8. Douglas Hung, D. C., "Enhancement and Feature Purification of Fingerprint Images", *Pattern Recognition*, vol. 26, no. 11, pp. 1661-1771, 1993.

9. 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.
10. Fingerprint Verification Competition, 2000, http://bias.csr.unibo.it/fvc2000/download.asp.
11. Galton, F., "Fingerprints", *London: Macmillan*, 1892.
12. Haralick, R., "Ridges and Valleys on Digital Images", *Computer Vision Graphics Image Processing*, vol. 22, pp. 28-38, 1983.
13. Hirata, T., and Katoh, T., "An Algorithm for Euclidean distance transformation", *SIGAL Technical Report of IPS of Japan*, 94-AL-41-4, pp. 25-31, September, 1994.
14. Hollingum, J., "Automated Fingerprint Analysis Offers Fast Verification", *Sensor Review*, vol. 12, no. 13, pp. 12-15, 1992.
15. Hong, L., Wan, Y., and Jain, A. K., "Fingerprint Image Enhancement: Algorithm and Performance Evaluation" , *IEEE Trans. PAMI*, vol. 20, no. 8, pp. 777-789, 1998.
16. Jain, A. K., Hong, L., and Bolle, R., "On-Line Fingerprint Verification" , *IEEE Trans. PAMI*, vol. 19, no. 4, pp. 302-314, 1997.
17. 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, 1997.
18. 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, 2000.
19. Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S., *Handbook of Fingerprint Recognition*, Springer-Verlag, New York, 2003.
20. Maio, D. and Maltoni, D., "Direct Gray-Scale Minutiae Detection In Fingerprints", *IEEE Trans. PAMI*, vol. 19, no. 1, pp. 27-39, 1997.
21. Mehtre, B. M. and Chatterjee, B., "Segmentation of Fingerprint Images - A Composite Method", *Pattern Recognition*, vol. 22, pp. 381-385, 1989.
22. Mehtre, B. M., "Fingerprint Image Analysis for Automatic Identification", *Machine Vision and Applications*, vol. 6, no. 2, pp. 124-139, 1993.
23. Moayer, B. and Fu, K., "A Tree System Approach for Fingerprint Pattern Recognition", *IEEE Trans. PAMI*, vol. 8, no. 3, pp. 376-388, 1986.
24. O'Gorman. L. and Nickerson, J. V., "An Approach to Fingerprint Filter Design", *Pattern Recognition*, vol. 22, pp. 29-38, 1989.
25. Ratha N. K., Chen, S. Y., and Jain, A. K., "Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images", *Pattern Recognition*, vol. 28, no. 11, pp. 1657-1672, 1995.
26. Rosenfeld, A. and Kak, A. C., *Digital Image Processing*, vol. 2, Academic Press Inc., Orlando, Florida, 1982.
27. Senior A., "A Combination Fingerprint Classifier", *IEEE Trans. PAMI*, vol. 23, no. 10, pp. 1165-1174, 2001.
28. Watson, C. I., Wilson, C. L., *Fingerprint Database*, National Institute of Standards and Technology, Special Database 4, FPDB, April, 1992.
29. Wegstein, J. H., "An Automated Fingerprint Identification System", *US Government Publication*, Washington, 1982.
30. Xiao, Q., and Raafat, H., "Fingerprint Image Post-Processing: A Combined Statistical and Structural Approach", *Pattern Recognition*, vol. 24, no. 10, pp. 985-992, 1991.
31. Shih, F. Y. and Pu, C. C.,"A Skeletonization Algorithm by Maxima Tracking on Euclidean Distance Transform", *Pattern Recognition*, vol. 28, no. 3, pp. 331-341, March 1995.