Computer Technology and Application 3 (2012) 401-407

A Block Processing Approach to Fingerprint Ridge-Orientation Estimation

Iwasokun Gabriel Babatunde, Akinyokun Oluwole Charles and Olabode Olatubosun

Department of Computer Science, Federal University of Technology, Akure, Nigeria

Received: April 14, 2012 / Accepted: May 05, 2012 / Published: June 25, 2012.

Abstract: Fingerprint is one of the most universally accepted biometrics used in different and relevant fields of man's endeavours including business transactions and human security. They have been used for the implementation of series of Automatic Fingerprint Identification Systems (AFIS) that have proved very adequate and efficient. Building an AFIS requires the implementation of different algorithms. One of such algorithms is the one concerned with fingerprint image enhancement which involves segmentation, normalization, ridge orientation estimation, ridge frequency estimation, filtering, binarization and thinning. In this research, the implementation of a modified approach to an existing ridge orientation estimation algorithm is presented with a view to increase speed and accuracy. The implementation was carried out in an environment characterized by Window Vista Home Basic operating system as platform and Matrix Laboratory (MatLab) as frontend engine. Synthetic images as well as real fingerprints obtained from selected staff and students of The Federal University of Technology, Akure (FUTA), Nigeria and the standard FVC2000 fingerprint database DB2 were used to test the adequacy of the resulting algorithm. The results show that the modified algorithm estimated the orientation with significant improvement over the original version.

Key words: Fingerprint, AFIS (automatic fingerprint identification systems), ridge orientation, biometrics, singular point.

1. Introduction

Fingerprint has continued to enjoy international recognition and approval in human verification and identification. It has continued to show superiority over other biometrics like face, iris, voice, gait, palm, hand, signature, middle ware and so on [1]. Fingerprints exist in different patterns ranging from left loop, right loop, whorl, arch and tented arch as shown in Fig. 1 [2]. In the loop pattern, the ridges enter from either side, re-curve round the core point (which is the point of maximum orientation) and pass out or tend to pass out the same side they entered. In the right loop pattern, the ridges enter from the right side while they enter from the left side in the left loop. In a whorl pattern, the ridges are usually circular round the core point while in the arch pattern, the ridges enter from one side, make a rise round the core point and exit generally on the opposite side.

The main components of any fingerprint used for identification and security control are the features it possesses. The features exhibit uniqueness defined by type, position and orientation from fingerprint to fingerprint and they are classified into global and local features [3-6].

Global features are those characteristics of the fingerprint that could be seen with the naked eye. They are the features that are characterized by the attributes that capture the global spatial relationships of a fingerprint. Global features include ridge pattern, type, orientation, spatial frequency, curvature, position and

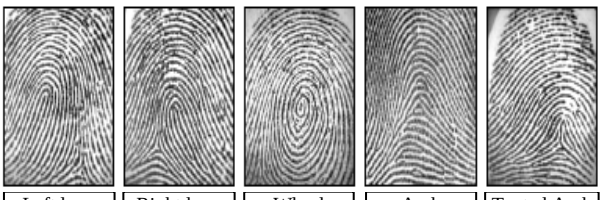

Fig. 1 Basic types of thumbprint pattern.

Corresponding author: Iwasokun Gabriel Babatunde, Ph.D., research field: pattern recognition and matching, factor analysis. E-mail: maxtunde@yahoo.com.

count. Others are type lines, core and delta areas. The local features are also known as minutiae points. They are the tiny, unique characteristics of fingerprint ridges that are used for positive identifications. Local features contain the information that is in a local area only and invariant with respect to global transformation.

Reliable and sound verification of fingerprints in any AFIS is always preceded with a proper detection and extraction of its features. A fingerprint image is firstly enhanced before the features contained in it could be detected or extracted. A well enhanced image will provide a clear separation between the valid and spurious features. Spurious features are those minutiae points that are created due to noise or artifacts and they are not actually part of the fingerprint [6-8]. This paper presents a practical discussion on the fingerprint ridge orientation estimation which is a very important part of the enhancement process. Section 2 presents a clear discussion on the modified fingerprint ridge orientation estimation algorithm while section 3 focuses on findings from conducted experiments. The conclusions are presented in section 4.

2. Fingerprint Ridge Orientation Estimation

In every fingerprint image, the ridges form patterns that flow in different directions. The orientations of ridges at locations A(x, y), and B(x, y) shown in Fig. 2 are the directions of the flow over a range of pixels.

The sequence of activities involved in fingerprint ridge orientation estimation is presented in Fig. 3.

There are two regions that describe any fingerprint image: namely the foreground region and the background region. The foreground regions contain the ridges and valleys. The ridges are the raised and dark regions of a fingerprint image while the valleys are the low and white regions between the ridges. The foreground regions often referred to as the Region of Interest (RoI) is shown for the image presented in Fig. 4. The background regions are mostly the outside regions where the noises introduced into the image during enrolment are mostly found. The essence of

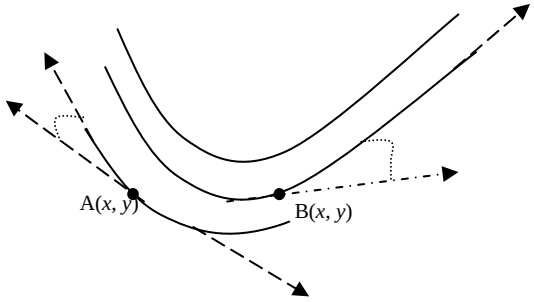

Fig. 2 The orientation of ridge pixels in a fingerprint.

Fig. 3 Sequence of activities for fingerprint ridge orientation estimation.

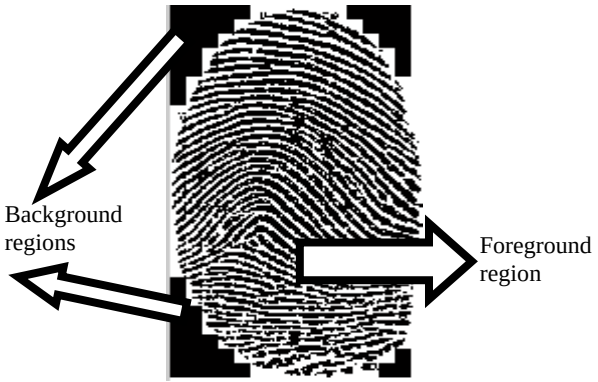

Fig. 4 A fingerprint image and its foreground and background regions.

segmentation is to reduce the burden associated with image enhancement by ensuring that focus is only on the foreground regions.

Normalization on its own is performed on the segmented fingerprint ridge structure for the standardization of the level of variations in the image grey-level values. By normalization, the grey-level values are made to fall within certain range good enough for improved image contrast and brightness. The modified fingerprint ridge segmentation and normalization algorithms implemented in Ref. [8] were adopted for this research. The algorithms sufficiently and effectively separate the foregrounds from the backgrounds using variance threshold approach. The segmented images were normalized to improve on their ridges and contrasts.

A modified version of the fingerprint ridge

orientation estimation algorithm proposed in Refs. [9-10] was implemented in this research. The modified algorithm is in the following phases:

(1) Firstly, blocks of size $S \times S$ were formed on the normalized fingerprint image.

(2) For each pixel, (p, q) in each block, the gradients $\partial_x(p, q)$ and $\partial_y(p, q)$ were computed as the gradient magnitudes in the *x* and *y* directions, respectively. $\partial_x(p, q)$ was computed using the horizontal Sobel operator while $\partial_y(p, q)$ was computed using the vertical Sobel operator [8].

(3) The local orientation of each pixel in a fingerprint was computed using its $S \times S$ neighborhood in Ref. [9]. However, in this research, each image is firstly divided into $S \times S$ blocks and the local orientation for each block centered at pixel $I(i, j)$ was then computed from

$$V_x(i,j) = \sum_{p=a}^{b}\sum_{q=c}^{d} 2\partial_x(p,q)\partial_y(p,q) \quad (1)$$

$$V_y(i,j) = \sum_{p=a}^{b}\sum_{q=c}^{d} \partial_x^2(p,q) - \partial_y^2(p,q) \quad (2)$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\frac{V_y(i,j)}{V_x(i,j)} \quad (3)$$

$a = i - \frac{S}{2}$, $b = i + \frac{S}{2}$, $c = j - \frac{S}{2}$, $d = j + \frac{S}{2}$ and $\theta(i,j)$ is the least square estimate of the local orientation of the block centered at pixel (i, j).

(4) The orientation image is then converted into a continuous vector field defined by

$$\varphi_x(i,j) = 2\cos^2\big(\theta(i,j)\big) \quad (4)$$

$$\varphi_y(i,j) = 2\sin\big(\theta(i,j)\big)\cos\big(\theta(i,j)\big) \quad (5)$$

φ_x and φ_y are the *x* and *y* components of the vector field, respectively.

(5) Gaussian smoothing is then performed on the vector field as follows:

$$\varphi'_X(i,j) = \sum_{p=-\vartheta}^{\vartheta}\sum_{q=-\vartheta}^{\vartheta} G(p,q)\varphi_x(i - ps, j - qs) \quad (6)$$

$$\varphi'_y(i,j) = \sum_{p=-\vartheta}^{\vartheta}\sum_{q=-\vartheta}^{\vartheta} G(p,q)\varphi_y(i - ps, j - qs) \quad (7)$$

$$\vartheta = \frac{S_\varphi}{2} \quad (8)$$

G is a Gaussian low-pass filter of size $S_\varphi \times S_\varphi$.

(6) The orientation field *O* of the block centered at pixel (i, j) is finally smoothed using the equation:

$$O(i,j) = 0.5\cos\left(\left(\left(\varphi'_y(i,j)\right) * \left(\varphi'_x(i,j)\right)^{-1}\right) * \sin\left(\left(\varphi'_y(i,j)\right) * \left(\varphi'_x(i,j)\right)^{-1}\right)\right) \quad (9)$$

3. Experimental Results

The implementation of the modified algorithm was carried out using Matlab version 7-6 on window Vista Home Basic Operating System. The experiments were performed on a Pentium 4-1.87 GHz processor with 1024MB of RAM. The purpose of the orientation estimation experiments is to analyze the performance of the modified algorithm under different conditions of images as well as generate the metrics that could serve the basis for the comparison of the results from the research with results from related works. Three sets of experiment were conducted for the performance analysis. The first set of experiments was on synthetic images. The *circsine* function [11] was employed for generating the synthetic images. The major arguments passed into the *circsine* function include a number for the size of the square image to be produced, a number for the wavelength in pixels of the sine wave and an optional number specifying the standard pattern of behaviour to use in calculating the radius from the centre. This defaults to 2, resulting in a circular pattern while large values give a square pattern. Where necessary, the MATLAB *imnoise* function was also used to generate noise and artifacts on synthetic images. The arguments passed into the *imnoise* function include the image on which noise is to be generated, noise type and noise level. The second set of experiments was on the fingerprint images obtained from selected persons in The Federal University of Technology, Akure, Nigeria while the third set of experiments was on FVC2000 fingerprint database DB2 (www.bias.csr.unibo.it/fvc2000/download.asp). This database contains benchmark fingerprints jointly produced by The Biometric Systems Laboratory,

Bologna, Pattern Recognition and Image Processing Laboratory, Michigan and the Biometric Test Center, San Jose, United States of America.

Figs. 5a-5c are synthetic images of size 410 × 410 and wavelength 15. They were obtained with the *imnoise* function using the salt and pepper noise level of 0, 0.2 and 0.3 respectively. The results of the ridge orientation estimation experiments on each of the three images are shown in Figs. 5d-5f respectively. These results show that for noise levels of 0 and 0.2, the modified ridge orientation algorithm effectively generated ridge orientation estimates that are very close to the actual orientations. However, for image with noise level of 0.3, the result shows a substantial number of ridge orientation estimates that significantly differ from the actual orientations. These results show that the performance of the modified algorithm depends on the image noise level. When the noise level on the image is within reasonable range, the algorithm does well while it fails when the noise level rises beyond the threshold which was found to be 0.29.

The orientation fields for the real fingerprint images were obtained around their singular points since they are prominent features used by any AFIS for fingerprint classification and matching. Parts of good quality images obtained from selected staff and students of FUTA are shown in Figs. 6a-6c. Their orientation estimates are shown in Figs. 6e-6g respectively. At the singular points, the orientation field is discontinuous and unlike the normal ridge flow pattern, the ridge orientation varies significantly. From these results, it is observed that there exists no deviation between the actual fingerprint ridge orientation and the estimated orientation of the vectors. In both cases, the algorithm produces accurate estimates of the orientation vectors such that they flow smoothly and consistently with the direction of the ridge structures in the images. In the superimposed version of images in Figs. 6e-6h, the contrast of the original image is lowered in each case. With the motive of improving the visibility of the orientation vectors

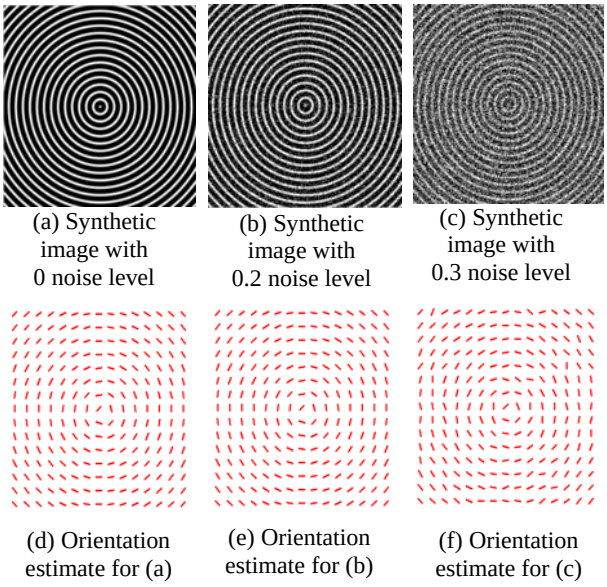

Fig. 5 Orientation and ridge frequency estimates for synthetic images of different noise levels.

against the background. The ridge orientation estimate for poor quality image obtained from a selected person in FUTA shown in Fig. 6d is presented in Fig. 6h. The estimate indicates a fairly smooth orientation field in some well defined regions while it shows failed results in areas of very poor quality as evident in the top-left and bottom regions of the estimate.

Parts of different qualities fingerprints contained in FVC2000 DB2 are shown in Figs. 7a-7d. This is a fingerprint database formulated through collaborative efforts and the sole aim of conducting experiment on it is for standard comparative analysis. The orientation estimates of the images shown in Figs. 7a-7d are shown in Figs. 7e-7h respectively. These results show that the orientations were perfectly estimated around the singular points by the modified algorithm in line with the qualities of the images in the same manner as images obtained from selected persons in FUTA.

The efficiency of the ridge orientation estimation algorithm was quantitatively measured by estimating the Mean Square Error (MSE) which represents the difference between the estimated and actual ridge orientation values in radians. Mean square error estimation results for the synthetic image shown in Fig. 5a under different conditions of noise for both the

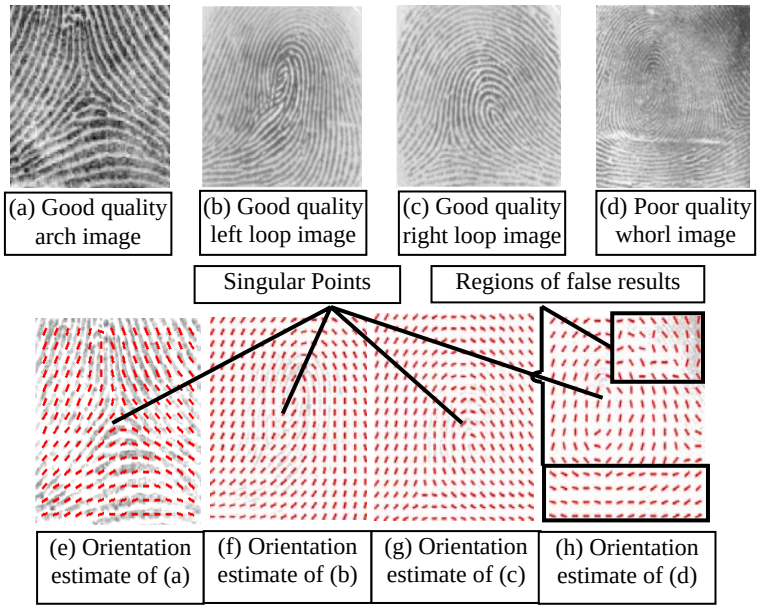

Fig. 6 Fingerprint images and their orientation estimates.

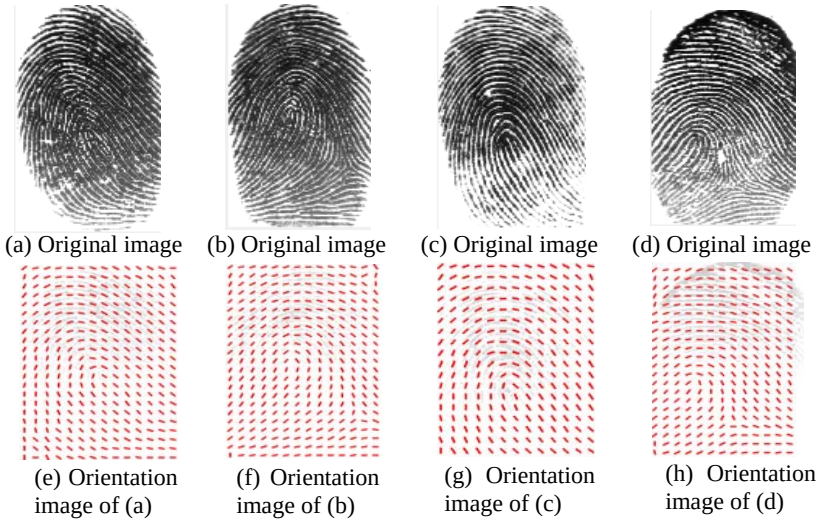

Fig. 7 Images from FVC2004 DB3 set 3 and their orientation estimates.

pixel processing approach in Ref. [10] and the block processing approach formulated for this research are shown in Table 1. The increasing mean square errors in both cases indicate that the accuracy of the two approaches decreases with increase in the noise level. It is also revealed that the orientation estimate is closer to the actual value for the block processing approach at any noise level. With lower standard deviation of 0.033473 for its MSE values, it is established that the block processing approach performs better than the pixel processing approach with MSE values of standard deviation of 0.143313. This higher performance is attributed to the fact that in the block processing approach, the degree of variation in the orientation estimates for pixels in a block is reduced to zero as each pixel assumes the orientation estimate for the center pixel of its host block. This significantly increases the ability of the algorithm to estimate the ridge orientation close to the actual value.

Table 2 shows the results of the quantitative measures of the average orientation estimation time for the images contained in FVC2000 fingerprint database

Table 1 Comparison of mean square error for ridge orientation estimates.

Noise level	Mean square error	
	R. Thai's study [10]	Current study
0.00	0.000255	0.000000
0.05	0.000765	0.000510
0.10	0.002720	0.000680
0.15	0.008670	0.001615
0.20	0.020910	0.001275
0.25	0.058735	0.002210
0.30	0.146370	0.034425
0.35	0.198050	0.044285
0.40	0.258485	0.056185
0.45	0.350540	0.068255
0.50	0.362270	0.092225

Table 2 Comparison of the computation time for the original and modified algorithm.

Dataset	Average computation time		Increase (%)
	Original algorithm	Modified algorithm	
FVC2000 DB2	1.89	1.14	39.68
106 images from FUTA	1.98	1.22	38.38

DB2 and those obtained from FUTA using the original and the modified algorithms under same conditions of coding and environments. Before the estimations, all the images in each case were converted to uniform sizes for equal number of pixels and blocks. It is revealed that the modified algorithm returned the orientation estimates at average times that are significantly lower than that of the original algorithm. This is attributed to the block processing approach that computes orientation estimate for only the center pixel in each block rather than for all the pixels. This leads to fewer computations and consequently, lesser computation time.

4. Conclusions

An implementation of a modified version of the fingerprint orientation algorithm proposed in Ref. [9] had been reported. The modification was done with a view to improve speed and performance. The comparison of the computation time of the original and modified versions under same conditions for selected fingerprint images shows that there is reduction in the completion time for the modified algorithm. While the pixel processing approach in the original algorithm subjects each pixel in the image to orientation estimation, the block processing approach firstly divides the image into $S \times S$ blocks and obtains the orientation estimate for the center pixels. This leads to reduction in the number of calculations and consequently reduced completion time. This also resulted in lesser number of computation and higher performance as attested to by the table of MSE for ridge orientation estimates.

References

[1] C. Roberts, Biometrics, available online at: http://www.ccip.govt.nz/newsroom/informoation-notes/2005/biometrics.pdf, accessed: July 2009.

[2] T.-Y. Jea, V. Govindaraju, A minutia-based partial fingerprint recognition system, Pattern Recognition 38 (2005) 1672-1684.

[3] C. Michael, E. Imwinkelried, Defence practice tips, a cautionary note about fingerprint analysis and reliance on digital technology, Public Defense Backup Centre Report, 2006.

[4] O.C. Akinyokun, C.O. Angaye, G.B. Iwasokun, A framework for fingerprint forensic, in: Proceeding of the First International Conference on Software Engineering and Intelligent System, Covenant University, Ota, Nigeria, 2010, pp. 183-200.

[5] O.C. Akinyokun, E.O. Adegbeyeni, Scientific evaluation of the process of scanning and forensic analysis of fingerprints on ballot papers, in: Proceedings of Academy of Legal, Ethical and Regulatory Issues, Vol. 13, No. 1, New Orleans, 2009.

[6] G.B. Iwasokun, O.C. Akinyokun, B.K. Alese, O. Olabode, Adaptive and faster approach to fingerprint minutiae extraction and validation, International Journal of Computer Science and Security 5 (4) (2011) 414-424.

[7] C. Sharat, C. Wu, V. Govindaraju, A systematic approach for feature extraction in fingerprint pattern recognition, Center for Unified Biometrics and Sensors (CUBS), University at Buffalo, NY, USA, 2004.

[8] G.B. Iwasokun, O.C. Akinyokun, B.K. Alese, O. Olabode, Fingerprint image enhancement: Segmentation to thinning, International Journal of Advanced Computer Science and Applications Indian 3 (1) (2012).

[9] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: Algorithm and performance evaluation, Pattern

Recognition and Image Processing Laboratory, Department of Computer Science, Michigan State University, 2006, pp. 1-30.

[10] R. Thai, Fingerprint image enhancement and minutiae extraction, Ph.D. Thesis, School of Computer Science and Software Engineering, University of Western Australia, 2003.

[11] P. Kovesi, MATLAB functions for computer vision and image analysis, School of Computer Science and Software Engineering, University of Western Australia, available online at: http:/www.cs.uwa.edu.au/~pk/Research/Matlab Fns/Index.html, accessed: February 20, 2010.