

## PAPER

# Orientation Field Estimation for Embedded Fingerprint Authentication System

Wei TANG<sup>†a)</sup>, *Nonmember*, Dongju LI<sup>†</sup>, Tsuyoshi ISSHIKI<sup>†</sup>, *Members*, and Hiroaki KUNIEDA<sup>†</sup>, *Fellow*

**SUMMARY** Orientation field (OF) estimation is a fundamental process in fingerprint authentication systems. In this paper, a novel binary pattern based low-cost OF estimation algorithm is proposed. The new method consists of two modules. The first is block-level orientation estimation and averaging in vector space by pixel level orientation statistics. The second is orientation quantization and smoothing. In the second module, the continuous orientation is quantized into fixed orientations with sufficient resolution (interval between fixed orientations). An effective smoothing scheme on the quantized orientation space is also proposed. The proposed algorithm is capable of stably processing poor-quality fingerprint images and is validated by tests conducted on an adaptive OF matching scheme. The proposed algorithm is also implemented into a fingerprint System on Chip (SoC) to confirm that it satisfies the strict requirements of embedded system.

**key words:** *orientation field, fingerprint authentication, feature extraction, binary pattern, image processing*

## 1. Introduction

The authentication of Personal identity by Automatic Fingerprint Identification System (AFIS) offers greater convenience and security than traditional password or ID card based methods since a password and ID card can easily be forgotten or stolen. A fingerprint is the pattern of ridges and valleys on the surface of the finger. The uniqueness of a fingerprint can be determined by the overall pattern of ridges and valleys as well as the local ridge minutiae (ridge ending and ridge bifurcation).

Most classical algorithms take the minutiae and the singular point (SP) as the distinctive features to represent the fingerprint in the matching process. Minutiae extraction mainly includes the following steps: Orientation Field (OF) Estimation, Ridge Enhancement, Binarization, Thinning, Minutiae Detection and post processing to handle the whole image [1]. SPs are directly extracted from OF. Thus, an algorithm to estimate OF accurately and robustly is desired as a fundamental process for the whole recognition system. Given the requirements of embedded systems, there are several problems to be solved in advance:

- Low workspace memory requirement for online processing;
- Low computational complexity in feature extraction and matching;

- Avoiding floating point calculation;
- Processing covers most parts of the fingerprint image.

Many algorithms have been proposed for OF estimation, such as gradient-based approaches [2]–[5], filter-bank based approaches [6], [7], methods based on high-frequency power in 3-D space [8], and 2-D spectral estimation methods [9]. However, it is reported that these methods do not provide results with the same accuracy as the gradient-based methods. This is mainly because most of them rely on a number of fixed possible templates or filters [1]. Furthermore, those methods usually employ floating point calculations, which makes the computation complexity too high for embedded system. Therefore, our motivation in this paper is to find a new method to estimate OF, which not only matches the strict requirements of embedded system but also performs as well as the classical methods. Given these requirements, a novel binary pattern based fingerprint OF estimation method is proposed. The new method consists of two modules. The first is block-level orientation estimation and averaging in vector space by pixel-level orientation statistics, as described in Sect. 2. In the second module presented in Sect. 3, the orientation field is quantized to reduce computational complexity. An effective smoothing scheme for the quantized orientation space is also proposed to fix the incorrect estimation in very noisy area where the averaging scheme in the first module fails. In Sect. 4, an adaptive OF matching scheme is introduced for large dataset evaluation of OF estimation algorithms. Finally, in Sect. 5, experiments are conducted to validate that the proposed algorithm is capable of stably processing poor-quality fingerprint images and satisfies the requirements of an embedded system [15].

## 2. Orientation Estimation in Vector Space

Before binarization, raw fingerprint image should be enhanced. Our enhance process includes a highpass filter to sharpen the ridge profile and a lowpass filter to remove the high frequency noise.

### 2.1 Ridge Orientation Analysis

In the foreground of a binarized fingerprint image, ridge and valley are labeled as white and black respectively. In a small block, the ridge profile is defined as the boundary pixels in the ridge which separate the ridge from the valley, as shown in Fig. 1. If the block size is small enough, the ridge profile

Manuscript received October 21, 2009.

Manuscript revised March 17, 2010.

<sup>†</sup>The authors are with the Dept. of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo, 152–8550 Japan.

a) E-mail: tangwei@vlsi.ss.titech.ac.jp

DOI: 10.1587/transinf.E93.D.1918

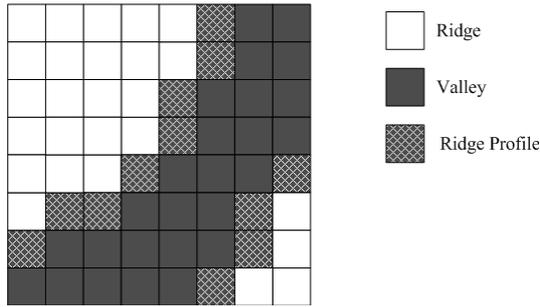


Fig. 1 Ridge profile in block 8 by 8 pixels.

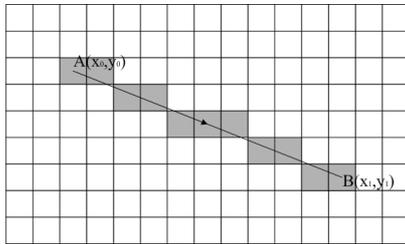


Fig. 2 An example of line represented in a binary image.

is an approximation of tangent of ridge curve. It is reasonable that by calculating the slope of the profile, the local ridge orientation<sup>†</sup> within the small block can be estimated. This idea is illustrated by a 3-step algorithm:

1. search the whole block and label the ridge profile;
2. find the start point  $(x_0, y_0)$  and end point  $(x_1, y_1)$  of ridge profile;
3. calculating  $\frac{y_1 - y_0}{x_1 - x_0}$ ,  $x_1 \neq x_0$ , otherwise the orientation is horizontal.

This algorithm is simple, but it is not so easy to search  $(x_0, y_0)$  and  $(x_1, y_1)$ . When the block of interest is large enough, it may include more than one identical shape of ridge profiles. Therefore, we propose an extended method to calculate the slope. Suppose a line is drawn in an image as shown in Fig. 2. The start point is  $A(x_0, y_0)$  and the end point is  $B(x_1, y_1)$ . The general formula for the line between two points  $A$  and  $B$  is given by:

$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0), \quad x_1 \neq x_0 \quad (1)$$

This formula gives the slope of the line  $AB$ , which is  $\frac{y_1 - y_0}{x_1 - x_0}$ . In another case, to obtain the slope of  $AB$ , one can calculate the accumulative steps along  $x$  and  $y$  directions, denoted by  $dx$  and  $dy$  respectively, from the ordered sequence of  $C$ , which is composed of all the points in line  $AB$ :

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \sum_{n=0}^{N-1} (x_{n+1} - x_n) \\ \sum_{n=0}^{N-1} (y_{n+1} - y_n) \end{pmatrix}, \quad x_{N-1} \neq x_0 \quad (2)$$

where  $N$  is the number of points in  $C$  and point  $(x_n, y_n) \in C$ . The slope will be given by  $\frac{dy}{dx}$ , which can also be represented by a vector  $\vec{O}(dx, dy)$ . As a result,  $dy$  will not be affected

Table 1 Five types of micropatterns and their contribution to the slope.

Type	Angle	Micropattern Configurations	Contribution to Slope
1	0°		$dx = dx \pm 1,$ $dy = dy$
2	45°		$dx = dx + 1,$ $dy = dy + 1$
3	90°		$dx = dx,$ $dy = dy + 1$
4	135°		$dx = dx - 1,$ $dy = dy + 1$
5	Don't care		-

by the calculation priority of  $y_{n+1} - y_n$  and  $y_{n+2} - y_{n+1}$ , etc (same situation for  $dx$ ). Thus, instead of constructing the sequenced set  $C$  within the block of interest, formula 2 can be replaced with statistical calculation of steps locally to avoid searching  $A$  and  $B$ .

In a local 2 by 2 window of binarized image in pixel level, there are  $2^4 = 16$  patterns. Excluding the configuration of all black and all white, the left 14 patterns can be categorized into 5 classes associating to their contribution to  $dx$  and  $dy$ , corresponding to 0°, 45°, 90°, 135°, and undefined respectively, as shown in the third column of Table 1. Because type 5 may be 45°, 135° or noise, this type is also discarded. It is easy to notice that pattern 1 represents the fact that the ridge flow steps one pixel horizontally from left to right (or from right to left, further discrimination is necessary for block orientation estimation, which is discussed in Sect. 2.2). Associated with the accumulative steps  $dx$  and  $dy$ , the contribution to the slope of each pattern is listed in the fourth column of Table 1. If we take the upper left pixel as interested, the pixel-level ridge orientation can be obtained by comparing the 2 by 2 window with the 16 configurations. By overlapping the 2 by 2 window, all the ridge profile pixels will be labeled by one identical value from type 1 to 4.

## 2.2 Block Level Ridge Orientation

To derive block level orientation from pixel-level orientation statistics, firstly, a histogram of four valid types of pixel orientation pattern is obtained and denoted as  $H(h_i | i = 1, 2, 3, 4)$ , where  $h_i \geq 0$ .  $h_i$  denotes the count of appeared type  $i$  pattern in a block size of  $w$  by  $w$ , and its corresponding angle is  $\theta_i = \frac{\pi \times (i-1)}{4}$ . If the block size is  $w$  by  $w$  pixels, there must be  $\sum h_i \leq w^2$ . Secondly, the accumulative steps  $dx$  and  $dy$  can be derived from histogram  $H$ . The only problem is to discriminate whether the slope vector is in the first or second quadrant. This can be solved by comparing the value of  $h_2$  and  $h_4$ . Therefore, there are 3 cases should be considered for generating the accumulative steps  $dx$  and  $dy$  except the case that all histogram values are 0. Here, all the possible cases are summarized and listed below:

<sup>†</sup>Ridge orientation represents a disorientated direction within the range of  $[0, \pi)$ .

1. if  $h_2 > h_4$

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} h_2 - h_4 + h_1 \\ h_2 - h_4 + h_3 \end{pmatrix} \quad (3)$$

2. if  $h_2 = h_4$

a. if  $h_1 \geq h_3$

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} h_1 \\ 0 \end{pmatrix} \quad (4)$$

b. if  $h_1 < h_3$

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} 0 \\ h_3 \end{pmatrix} \quad (5)$$

3. if  $h_2 < h_4$

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} h_2 - h_4 - h_1 \\ h_4 - h_2 + h_3 \end{pmatrix} \quad (6)$$

Thus, the orientation  $\theta \in [0, \pi)$  of this block can be derived from the vector  $\vec{O}(dx, dy)$ :

$$\theta = \begin{cases} \arctan\left(\frac{dy}{dx}\right) & dx \neq 0 \\ 0 & dy = 0 \\ \frac{\pi}{2} & dx = 0, dy \neq 0 \end{cases} \quad (7)$$

where  $\arctan$  is calculated by:

$$\arctan(z) = \frac{i}{2} \log\left(\frac{i+z}{i-z}\right) \quad (8)$$

### 2.3 Averaging the orientation vectors

The OF obtained from Sect. 2.2 is rough and not accurate, because the local window may be comparatively small in some area. Noise or broken ridge flow has the effect of false presentation of local orientation. An averaging scheme will help to correct by investigating a larger window.

To average the orientation vectors, a local 3 by 3 window lowpass filter for the pixel-level orientation histogram  $H_{mn}(h_{i_{mn}} | i = 1, 2, 3, 4)$  on block level is applied, where  $m$  and  $n$  denote the  $m$ -th column and  $n$ -th row of block wise OF. The parameter of filter mask LPM is listed in Table 2.

The center block with the heaviest weight is the block of interest. Thus, the output  $H'_{mn}(h'_{i_{mn}} | i = 1, 2, 3, 4)$  will be calculated by:

$$\begin{aligned} h'_{i_{mn}} &= \frac{\sum_{k=-1}^{k=1} \sum_{l=-1}^{l=1} h_{i_{(m+k)(n+l)}} \times LMP(k, l)}{\sum_{k=-1}^{k=1} \sum_{l=-1}^{l=1} LMP(k, l)} \\ &= \frac{\sum_{k=-1}^{k=1} \sum_{l=-1}^{l=1} h_{i_{(m+k)(n+l)}} \times LMP(k, l)}{16} \end{aligned} \quad (9)$$

The division by 16 can be implemented with a 4-bit shift operation. Actually, the division does not affect the result of  $\frac{dy}{dx}$  as defined in previous section and thus is omitted.

**Table 2** Parameter of lowpass filter window for orientation histogram of each block centered by OF (m,n).

1 (m-1,n-1)	2 (m-1,n)	1 (m-1,n+1)
2 (m,n-1)	4 (m,n)	2 (m,n+1)
1 (m+1,n-1)	2 (m+1,n)	1 (m+1,n+1)

### 3. Orientation Field Quantization and Smoothing

The resolution of the vector space OF estimated by the previous model is luxuriously high. The orientation  $\theta$  derived from the vector  $\vec{O}(dx, dy)$  by Formula 8 has implementation with fixed point arithmetic but the computation complexity is rather high. Moreover, noisy part will require larger window to average. Simply enlarging the window will degrade the resolution accuracy around SP, because high curvature area requires small block to satisfy the assumption that the ridge profile is the tangent of ridge curve. Quantizing the OF into a number of fixed orientations, which reserve sufficient resolution, can speed up the processing with maintaining the accuracy. Thus, in Sect. 3.1, the method which quantizes a continuous orientation into one of N predefined discrete orientations is presented. An hybrid smoothing algorithm in the quantized OF is also proposed, which can fix the incorrect estimation of the previous module, and will be described in Sect. 3.2.

#### 3.1 Quantization by Inner Product

To quantize the vector  $\vec{O}(dx, dy)$  into fixed orientations, a simple way is to construct the predefined orientations as vectors with the same amplitude. For example, in resolution of 4, the reference vector of quantized orientations of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  can be denoted by  $(256, 0)$ ,  $(181, 181)$ ,  $(0, 256)$  and  $(-181, 181)$  respectively. Then, by performing inner product between the reference vectors and a vector  $\vec{O}(dx, dy)$ :

$$IP(i) = x(i) \times dx + y(i) \times dy \quad (10)$$

where  $(x(i), y(i))$  denotes the  $i$ -th predefined orientation vector,  $IP(i)$  denotes the corresponding inner product. The expected quantized orientation for  $\vec{O}(dx, dy)$  is  $j$ , where  $IP(j) = \max(IP(0), \dots, IP(3))$ . This method avoids division and floating point calculation. However, the computation complexity increases with quantization resolution. Therefore, a suitable quantization resolution should be estimated for system optimization.

#### 3.2 Smoothing Quantized OF

The basic idea to smooth the quantized OF is based on the fact that the ridge flow is continuous, so that the tangent of the ridge may change from one in the neighbors, but may not jump from the one. In this section, 2 smoothing schemes are designed and combined to smooth the quantized OF efficiently and effectively.

##### 3.2.1 One Dimensional Lowpass Filtering

Let  $i, j$  denote the block of  $i$ -th column and  $j$ -th row in the OF. The idea of one dimensional lowpass filtering is to calculate the orientation difference between current block  $\theta_{ij}$  and its previous block  $\theta_{(i-1)j}$  and next block  $\theta_{(i+1)j}$  in the

case of horizontal scanning, named  $D_p$  and  $D_n$  respectively:

$$\begin{pmatrix} D_p \\ D_n \end{pmatrix}_{ij} = \begin{pmatrix} \text{Difference}(\theta_{ij}, \theta_{(i-1)j}, n) \\ \text{Difference}(\theta_{(i+1)j}, \theta_{ij}, n) \end{pmatrix} \quad (11)$$

where the difference (or distance, in the range of  $(-\frac{n}{2}, \frac{n}{2}]$ ) between two quantized angle  $a$  and  $b$  in the congruence class modulo  $n$  is calculated by:

$$\text{Difference}(a, b, n) = \begin{cases} a - b & \text{if } -\frac{n}{2} < a - b \leq \frac{n}{2}; \\ a - b + n & \text{if } a - b \leq -\frac{n}{2}; \\ a - b - n & \text{else.} \end{cases} \quad (12)$$

For any  $\theta_{ij}$ , the averaging result is given by:

$$\theta' = \begin{cases} \left( \theta + \frac{D_n - D_p}{2} \right) \bmod n & \text{if } D_p < 0, D_n \geq 0 \\ & \text{or } D_p \geq 0, D_n < 0; \\ \left( \theta + \frac{D_n + D_p}{2} \right) \bmod n & \text{else,} \end{cases} \quad (13)$$

where the subscript  $ij$  for  $\theta_{ij}$ ,  $\theta'_{ij}$ ,  $D_{n_{ij}}$  and  $D_{p_{ij}}$  is omitted for simplicity.

It is easy to be noticed that:

$$D_{p_{ij}} = D_{n_{(i-1)j}}. \quad (14)$$

Thus, for each row processed from left to right, it is simpler to generate all of values of  $D_p$  at first and then calculate the averaged orientation of each block by formula 13 using Eq. 14, which makes this one dimensional lowpass filter a nonrecursive one.

### 3.2.2 Two Dimensional Lowpass Filtering on Quantized OF

If a small block size, such as  $8 \times 8$  pixels, is selected to reserve sufficient resolution for high curvature area around SP, mutational orientations will appears in some area as group of blocks. The nonrecursive one dimensional lowpass filtering scheme proposed in Sect. 3.2.1 is not capable to produce correct estimation result for very noisy area, which generates mutational orientations. We herein propose a recursive 2 dimensional lowpass filtering scheme by searching the most possible orientation of the block of interest to solve the mutational orientation area problem. The idea is that if in a local window of a certain size, OF is smoothed to the perfect result, the variance of center block to all the other surrounding blocks is supposed to be the least. We will find the quantized orientation of the center block, which minimize the average difference from orientations of the surrounding blocks, by calculating all average differences for all the candidate orientations. Because the orientations are quantized into limited number of values, it makes the exhaustive searching possible.

Define the variance between an angle value  $m$ ,  $m \in [0, n - 1]$  and a local window of  $w$  by  $w$  blocks:

$$\text{VAR}_m = \sum_{k=-\frac{w}{2}}^{\frac{w}{2}} \sum_{l=-\frac{w}{2}}^{\frac{w}{2}} (\text{Difference}(m, \theta_{(i+k)(j+l)}, n))^2 \quad (15)$$

the result of averaging is  $q$ , where  $\text{VAR}_q = \min\{\text{VAR}_0, \text{VAR}_1, \dots, \text{VAR}_{n-1}\}$ . The filtering is processed from up to down, left to right on the quantized OF. The averaging result  $q$  is written into the current block of coarse OF and loaded when calculating the VARs for the next block. By this recursive scheme, the whole mutational orientation segment can be rectified, which is quite similar as an erosion process.

### 3.2.3 Hybrid Lowpass Filter

Compared to the two dimensional filter described in Sect. 3.2.2, one dimensional lowpass filter alleviates the computation complexity from  $O(n \times N \times M)$  to  $O(N)$ , where  $n$  is the quantization resolution,  $N$  is the number of orientation blocks to be processed and  $M$  is the local window size. In our implementation, the 8 neighboring blocks encircled the block of interest are selected as the local window. Therefore, the smoothing of quantized OF can be carried out effectively and efficiently by embedding the two dimensional filter into the one dimensional one through setting a threshold  $T$  for  $D_{p_{ij}}$ . If  $D_{p_{ij}}$  is sufficient large, two dimensional filter will be processed on current block  $\theta_{ij}$  and update  $D_{p_{ij}}$  in prior to the calculation of  $D_{p_{i(j+1)}}$ . For the block size of  $8 \times 8$  pixels, the empirical value for the threshold  $T$  becomes  $\frac{\pi}{6}$ . Because for most part of the OF,  $D_p$  is less than  $\frac{\pi}{6}$ , the computation complexity of the hybrid filter is approximately  $O(N)$ .

## 4. Adaptive OF Matching

Since there is no ground truth for OF of fingerprints, measurements of objective error are difficult to be constructed. Evaluation through human inspection is subjective and only executable for small data set. Some authors conducted objective measurement indirectly by investigating the results of sequent processing, for instance, the effect on SP detection, false minutiae detected, etc. However, the haleness of the sequent processing will confuse the evaluation of orientation estimation algorithms.

One way to construct an objective measurement is by measuring the difference ratio ( $DR$ ) between genuine OFs. The  $DR$  is defined as the average difference between all paired elements of 2 aligned OFs. A robust algorithm will produce similar OFs for genuine pairs which correspond to low  $DR$ . However, the  $DR$  distributions are difficult to be compared directly. Because the  $DR$  between an impostor OF pair is irrelevant to the accuracy of OF estimation algorithms, we plot the accumulative  $DR$  distributions of genuine and impostor against each other, which is actually the Receiver Operating Characteristic (ROC) curve of the OF based verification, to compare the performances of variant OF estimation algorithms. The  $DR$  becomes a representation of the similarity score. High (low)  $DR$  corresponds to low (high) similarity score.

Here, an adaptive OF matching scheme is introduced to solve the alignment in advance, along with the calculation of the  $DR$  to evaluate the robustness and stability of orientation estimation algorithm for large data set.

Although there will be no orientation in the background area, the orientation estimation algorithm will produce a orientation value as a result. This result will show high uncertainty among genuine images. Therefore, background area should be excluded from  $DR$  calculation. The background detection algorithm described in [14] uses the coherence, the mean, and the variance of the fingerprint image as features and morphological operators to smooth the detection result. Because the coherence can only be carried out by gradient-based method, in this paper, only the mean and the variance are adopted, along with the morphological operators, by the background detection algorithm used in experiment. This simplified background detection algorithm preserves sufficient accuracy and is independent to OF estimation algorithms.

The adaptive matching scheme works as following. For an alignment parameters  $(\delta x, \delta y)$  that are applied to align  $Q$  and  $P$ , where  $Q$  and  $P$  represents two OFs, it suggests that the block of the  $i$ -th row and  $j$ -th column in the foreground area of  $Q$  aligns with the block of the  $(i + \delta x)$ -th row and  $(j + \delta y)$ -th column in  $P$ . The alignment of the pair of orientation blocks is valid for  $DR$  calculation when the position  $(i + \delta x, j + \delta y)$  is also within the foreground area of the input. Suppose the valid aligned areas are denoted by  $A$  and  $A'$  for input and template, respectively, the  $DR$  is given by:

$$DR = \frac{\sum_{ij} (\text{Difference}(\theta_{ij}, \theta'_{i'j'}, n))^2}{N \times (\frac{n}{2})^2}, \theta_{ij} \in A, \theta'_{i'j'} \in A' \quad (16)$$

where  $N$  is the total number of aligned orientation blocks, and  $i' = i + \delta x, j' = j + \delta y$ . Here the range of  $DR$  calculated by Eq. 16 is  $[0, 1]$ . In the case of continuous OF, Eq. 16 can be applied by just replacing  $n$  with  $\pi$ . The actual  $DR$  is the minimum of  $DR$  values for all possible transformation parameters  $(\delta x, \delta y)$  between  $Q$  and  $P$ , with the restriction of  $N > \varepsilon$ , where “ $\varepsilon$ ” is the threshold for valid foreground area size. In our experiment, the optimal value of “ $\varepsilon$ ” is found to be approximately 1/3 of the total image size.

## 5. Experiment Results

In this section, some experiment results will be presented. First in Sect. 5.1, the processing results of several examples with different image quality are presented to show the robustness of the proposed algorithm. And in the following Sect. 5.2, the previously derived results are applied to a large number of fingerprints by the adaptive OF matching. The effect of quantization resolution on accuracy is also given. Finally, in Sect. 5.3, some computational aspects of the algorithms will be presented.

### 5.1 Small Data Set Evaluation

Figure 3 presents the processing results of each stage of different quality fingerprint images, including good, normal, poor and extremely poor quality. The description about

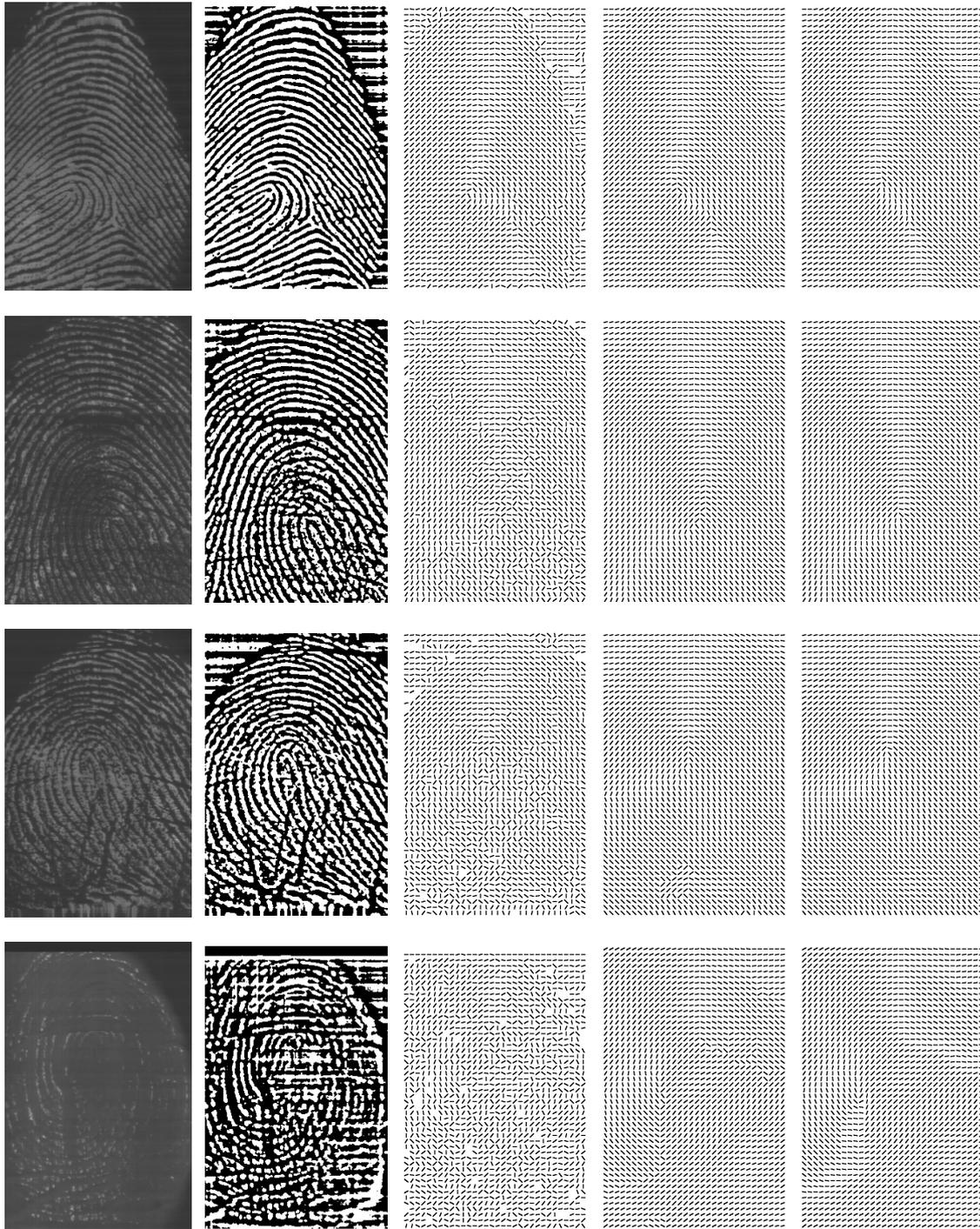
quality will be presented in Sect. 5.2.2. For the former 3 kinds of quality, the proposed algorithm produces smoothed OF. For the extremely poor-quality image, because of too many unrecoverable parts, although it can be classified as a whorl type fingerprint by human inspection, the proposed algorithm fails to produce an OF which describes a good whorl pattern. For the poor-quality image, because of the cutlines in the lower part, after local  $3 \times 3$  averaging, blocks in a small area is labeled with orientation around  $45^\circ$ . However, by human inspection, the orientation in this area should be around  $120^\circ$  to  $160^\circ$ . This small area is the so called mutational orientation area. As shown, the hybrid lowpass filter presented in Sect. 3.2.3 fixed this error. Furthermore, the 2 dimensional lowpass filter is robust in the high curvature area around SP, in which the difference between neighboring blocks may be larger than the threshold  $\frac{\pi}{6}$ .

Figure 4 shows the OF estimation results of the method proposed in [8]. It is shown that 8 fixed orientations can not present the continuous ridge flow accurately. The estimation error of this method is similar to the quantization error caused by quantizing a continuous OF into 8 fixed orientations. In the result of poor quality image, the mutational area is not correctly detected.

## 5.2 Experiment Conducted on Large Data Set

### 5.2.1 Experimental Database

To evaluate our algorithm by the adaptive OF matching scheme, a suitable database should be selected, where the translation and rotation effect should be limited. This is based on the consideration that in the case of translation, the adaptive OF matcher will only evaluate the robustness on the small overlapping area, other area can not be involved in the evaluation; in the case of rotation, the relative rotation coefficient should be estimated and introduced into the matcher, which makes the evaluation too complicated. To simplify the evaluation scheme, instead of using public data set such as FVC2004 [10] databases with extreme translation and rotation effect, we have run the experiments on our private database, which consists of 1240 images collected from 248 fingers of 80 persons (5 prints per finger), captured by a sweep sensor at 500 dpi (dot per inch) and categorized into 362, 413, 417 and 48 of good, normal, poor and extremely poor quality images by human inspection. Some examples of images with different subjective quality are shown in Fig. 3. For the good quality one, the ridge texture is clear with sufficient contrast against valley and continuous with few broken segment. For the normal quality one, the contrast of the ridge in some area is low, but it seems recoverable, there also will be several cutlines which produce disconnected ridges. For the poor quality one, the majority part of the foreground is suffered from broken ridge flows, cutlines and low contrast, very difficult to be recovered by local-processing enhancement methods. For the extremely poor quality, more than 2/3 of the foreground area is of low contrast, which is unrecoverable. This



**Fig. 3** Processing results of proposed algorithm on fingerprints of different quality. Each row shows good, normal, poor, extremely poor quality images, respectively; each column corresponds to original, binarized, coarse blockwise OF, local  $3 \times 3$  averaged OF and hybrid lowpass filtered OF after quantization with resolution 24, respectively.

sensor helps to restrict the user's behavior to eliminate translation and rotation effect. Therefore, for each run the cross-match process will generate  $5 \times 5 \times 248/2 = 3,100$  genuine and  $1240 \times 248 \times 5/2 = 768,800$  impostor attempts. The image size is restricted to  $256 \times 400$  pixels and the window size for orientation estimation is 8 by 8 pixels. Thus, an OF

of 32 by 50 blocks will be extracted by each method.

### 5.2.2 Background Detection Results

As mentioned in Sect. 4, the background area should be excluded for matching. Here in Fig. 5, the background detec-

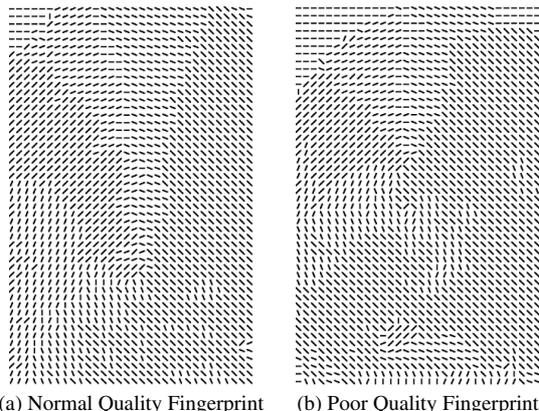


Fig. 4 OF estimation results of low orientation resolution (8 fixed orientation mask) method.



Fig. 5 Background detection results of good, normal, poor and extremely poor quality fingerprints, black area represents the background, grey area represents the foreground.

tion results are shown for the 4 images listed in Fig. 3. It is shown that although the coherence information is not included for background detection, the results show that sufficient accuracy is preserved. The experiment result shows that for the extremely poor quality fingerprint, the whole image has no foreground. Actually, by tuning the threshold for mean and variance, there will be some small segments of foreground detected. However, after morphological operation, these foreground area will be removed because they are too small and unconnected. As for the adaptive matching, the *DR* of two OFs estimated by a certain algorithm is calculated by those paired blocks both of which are not background.

5.2.3 Evaluation Criteria

False Acceptance Rate (FAR) and False Reject Rate (FRR) are two important error rates, which estimate the performance of a fingerprint verification system at various thresholds. In our case, they correspond to the accumulative *DR* distribution of impostor and genuine OFs matching. Receiver Operating Characteristic (ROC) curve plots the FRR against the FAR at different *DR* thresholds. Equal Error Rate (EER), which represent the thresholds where  $FRR(DR) = FAR(DR)$  in the ROC curve is also an interested criteria. For un-intersected ROC curves, only by comparing the EER values, the one with the better performance can be pointed out.

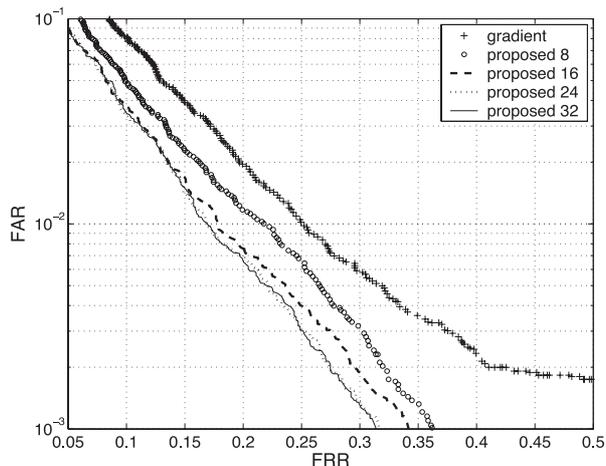


Fig. 6 Results of adaptive OF matching of different OF estimation methods. The methods include (1) gradient: the gradient based method; (2) proposed 8, 16, 24 and 32: the proposed method with quantization resolution 8, 16, 24 and 32 respectively.

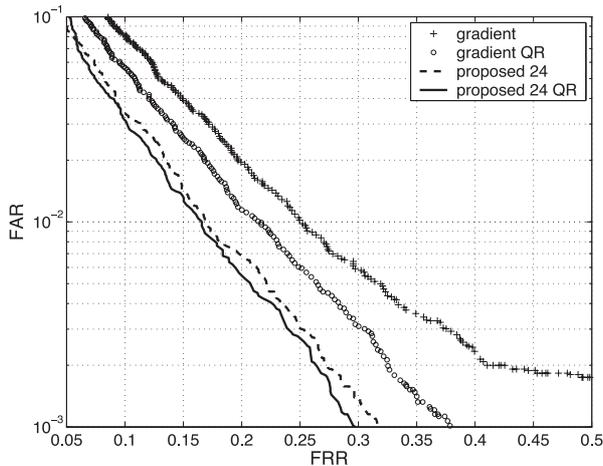
Table 3 EERs of proposed method with different quantization resolution and the gradient based method.

Methods	EER (%)
Proposed 8	7.395
Proposed 16	6.7744
Proposed 24	6.6001
Proposed 32	6.6628
Gradient-based	9.021

5.2.4 Evaluation Results

As explained in Sect. 1, the aforementioned OF estimation methods do not provide as much accurate results as the gradient-based method. One reason is fixed number of orientations as shown in Fig. 4 do not provide as much accuracy of ridge flow representation as the continues one. Another reason is the alternative pixel orientation estimation in gray level [9] does not enroll extra information and thus only increasing the computational complexity without achieving more accuracy. Therefore, in the large set evaluation, we only compare the proposed method with the gradient-based method.

As shown in Fig. 6, on our private database, the proposed method which consists of the two modules with quantization resolution of more than 16 produced similar results. This is because fixed window size limits the orientation resolution. As the resolution decreased from 16 to 8, the verification result became worse, which is because lower resolution will lose the accuracy of ridge flow representation. The EERs produced by the gradient method [4] and the proposed method with different resolution are listed in Table 3. If the 465 poor and extremely poor quality images are excluded, the EER of the proposed algorithm with the quantization resolution 24 improves from 6.60% to 6.32%. While the gradient based method improves significantly from 9.02% to 7.68%. Same conclusion can be drawn by examining the



**Fig. 7** Effect of excluding poor and extremely poor quality images. QR stands for quality restriction.

ROC curves shown in Fig. 7. This means the proposed algorithm is more stable and robust to process the poor-quality images than the gradient based method. The results of gradient method and the proposed method with quantization resolution of 24 are plotted twice in Fig. 6 and Fig. 7 for a clearer comparison visualization.

### 5.3 Computational Aspects

From multirate signal processing, it is known that the filtering and decimation steps can be implemented very efficiently using polyphase filters by interchanging the order of decimation and filtering [16]. Using this method, an efficient and optimized C implementation, the calculation of  $8 \times 8$  block OF requires less than 3 k words working memory, which benefits from that the processing is based on binarized image. Because the processing is based on gray scale image, other 3 listed methods require much more working memory. Filter-bank based method processes 2-D gabor filter on local blocks. High-frequency power based methods uses pre-defined mask for matching to detect the orientation which achieves the best enhancement for a local ridge. Although fixed-point solution can be designed, both methods require high computational complexity and result in much longer processing time. An overall comparison of processing times and working memory between the proposed method and other methods is given in Table 4. It is shown in the previous section that the proposed algorithm with quantization resolution 24 and 32 produces very similar verification results for both ROC curves and EERs. Considering the computational complexity for quantization, a resolution of 24 is suitable for embedded system implementation.

In our previous work [15], a fingerprint system-on-chip (SoC) with bit serial FPGA engine was proposed. The system chip includes a 64 KB ROM in which algorithms such as fingerprint image processing and minutiae extraction are embedded. A 32-bit RISC processor was used in the system.

**Table 4** Comparison of OF estimation methods. Platform: Pentium 4 2.8 GHz CPU, 1 GB RAM.

Methods	Time (ms)	Memory (words)
Proposed 8	2.7	3 k
Proposed 16	2.8	3 k
Proposed 24	3.1	3 k
Proposed 32	3.4	3 k
Gradient-based	8.7	20 k
Filter-bank based	89.4	48 k
High-frequency power in 3-D space	103.5	52 k

The processor works in 200 MHz frequency with 8 KB data cache, 8 KB instruction cache and memory protection unit. Therefore, on the above mentioned hardware environment, processing carried out by the proposed algorithm will be always executed in the data cache of the processor. In the case of image size with  $400 \times 256$  pixels, it only takes about 14.4 ms to calculate the  $8 \times 8$  blockwise OF with a resolution of 24.

## 6. Conclusion

In this paper, a novel binary pattern based low-cost OF estimation algorithm is proposed. The new method consists of two modules. The first is block-level orientation estimation and averaging in vector space by pixel-level orientation statistics. The second is orientation quantization and smoothing. In this module, the continuous orientations are quantized into several fixed orientations with sufficient resolution and smoothed by a hybrid lowpass filter. The proposed algorithm is capable of stably processing low-quality fingerprint images and satisfying the strict requirements of an embedded system.

## References

- [1] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.
- [2] A. Jain and L. Hong, "On-line fingerprint verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, no.4, pp.302–314, 1997.
- [3] A. Jain, L. Hong, S. Pankanti, and R. Bolle, "Identity authentication using fingerprints," *Proc. IEEE*, vol.85, no.9, pp.1365–1388, 1997.
- [4] A.M. Bazen and S.H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, no.7, pp.905–919, 2002.
- [5] A.K. Jain, S. Prabhakar, L. Hong, and S. Parkarti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Process.*, vol.9, no.5, pp.846–859, 2000.
- [6] A. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.21, no.4, pp.348–359, 1999.
- [7] K. Karu and A.K. Jain, "Fingerprint classification," *Pattern Recognit.*, vol.17, no.3, pp.389–404, 1996.
- [8] L. O'gorman and J.V. Nickerson, "An approach to fingerprint filter design," *Pattern Recognit.*, vol.22, no.1, pp.29–38, 1989.
- [9] C.L. Wilson, G.T. Candela, and C.I. Watson, "Neural network fingerprint classification," *J. Artif. Neural Network*, vol.1, no.2, pp.203–228, 1994.
- [10] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, "FVC2004: Third Fingerprint Verification Competition," *Proc. In-*

ternational Conference on Biometric Authentication (ICBA), pp.1-7, Hong Kong, July 2004.

- [11] S. Wang, W.W. Zhang, and Y.S. Wang, "Fingerprint classification by directional fields," Proc. Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02), Pittsburgh, 2002.
- [12] J. Boer, A. Bazen, and S. Cerez, "Indexing fingerprint database based on multiple feature," Proc. ProRISC 2001 Workshop on Circuits, Systems and Signal Processing, 2001.
- [13] A.M. Bazen and R.N.J. Veldhuis, "Likelihood-ratio-based biometric verification," IEEE Trans. Circuits Syst. Video Technol., vol.14, no.1, pp.86-94, Jan. 2004.
- [14] A.M. Bazen and S.H. Gerez, "Segmentation of fingerprint images," 12th Ann. Workshop Circuits, Systems and Signal Processing, Nov. 2001.
- [15] Y. Wang, D. Li, T. Isshiki, and H. Kunieda, "A novel fingerprint SoC with bit serial FPGA engine," IPSJ Digital Courier, vol.1, pp.226-233, 2005.
- [16] J.G. Proakis, C.M. Rader, F. Ling, and C.L. Nikias, Advanced Digital Signal Processing, Macmillan Publishing Company, New York 1992.



**Hiroaki Kunieda** received B.E., M.E. and D.E. degrees from Tokyo Institute of Technology in 1973, 1975 and 1978, respectively. He was Research Associate in 1978 and Associate Professor in 1985 at Tokyo Institute of Technology. He is now a Professor at Dept. of Communications and Integrated Systems in Tokyo Institute of Technology. He has been engaged in researches on distribute circus, switched capacitor circus, IC circus simulation, VLSI CAD, VLSI signal processing and VLSI design. His current

research focus on VLSI multimedia processing, design for system-on-chip, VLSI signal processing, VLSI architecture, and VLSI CAD. He is a member of IEEE CAS & SP society and IPSJ.



**Wei Tang** received the B.S. degree from Dept. of Computer Science, Wuhan University of Science and Technology, and M.E. degree from School of Software Engineering, Wuhan University, China, in 2001 and 2004 respectively. He is now pursuing Ph.D. degree in Dept. of Communications and Integrated Systems at Tokyo Institute of Technology. His research interests are pattern recognition, image processing and image database indexing.



**Dongju Li** received B.S. degree from Liaoning University and M.E. degree from Harbin Institute of Technology, China, in 1984 and 1987, respectively. She worked as an IC design engineer in VLSI Design Laboratory of Northeast Micro-electronics Institute, Electronic Industry Bureau, China, from 1987-1993. She received Ph.D. degree from Tokyo Institute of Technology in 1998. She is now a Research Associate at Dept. of Communications and Integrated Systems at Tokyo Institute of Technol-

ogy. Her research interests are multiprocessor architecture and VLSI design.



**Tsuyoshi Isshiki** received the B.E. and M.E. degrees in electrical and electronics engineering from Tokyo Institute of Technology in 1990 and 1992, respectively. He received his Ph.D. degree in computer engineering from University of California at Santa Cruz in 1996. He is now an Assistant Professor at Department of Communications and Integrated Systems in Tokyo Institute of Technology. His research interests include high-level design methodology for configurable systems, bit-serial synthesis, FPGA architecture, image processing, computer graphics and speech synthesis.

architecture, image processing, computer graphics and speech synthesis.