Fragile Bits in Palmprint Recognition

Lin Zhang, *Member, IEEE*, Hongyu Li, and Junyu Niu

***Abstract*—Recent years have witnessed a growing interest in developing automatic palmprint recognition methods. Among them, coding-based ones, representing the texture of a palmprint using a binary code, are most prevalent and successful. We find that not all bits in a code map generated by a specific coding scheme are equally consistent. A bit is deemed *fragile* if its value changes across code maps created from different images of the same palmprint. In this paper, we first analyze the fragile bits phenomenon in a state-of-the-art palmprint coding scheme, namely, binary orientation co-occurrence vector (BOCV). Then, based on our analysis, we extend BOCV to E-BOCV by incorporating fragile bits information in appropriate ways. Experiments conducted on the benchmark dataset demonstrate that E-BOCV can achieve the highest verification accuracy among all the state-of-the-art palmprint verification methods evaluated. To our knowledge, this is the first work investigating the fragile bits of coding-based palmprint recognition approaches.**

***Index Terms*—BOCV, fragile bits, palmprint.**

I. Introduction

BOLSTERED by the requirements of numerous applications, such as access control, aviation security, or e-banking, recognizing the identity of a person with high confidence has become a topic of intense study. Biometrics based methods are drawing more and more attention. As an important member of the biometrics family, palmprint-based personal authentication systems have been corroborated to have the merits of high distinctiveness, robustness, high user-friendliness, and cost effectiveness [1], [2]. In the past decade or so, a plethora of algorithms have been proposed for palmprint recognition [3], [4].

Among various palmprint recognition schemes, coding-based ones are the most prevalent and appealing since in general they have the advantages of high accuracy, robustness to illumination variation, and fast feature extraction and matching speed. In a typical coding-based method, each field of the code map is assigned a bit-wised code, based on the quantization result of the image's responses to a set of specific filters. PalmCode proposed in [1] uses a single Gabor filter to extract the local phase information. Its computational architecture is the same as the IrisCode [5]. In [6], Kong and Zhang proposed the competitive code (CompCode) scheme, which encodes the local orientation field of a palmprint using symmetric Gabor filters along six different orientations. In [7], Jia *et al.* proposed a different coding method to extract the local orientation information of palmprints, namely robust line orientation code (RLOC), which is based on a modified finite Radon transform. In [8], Sun *et al.* used differences between two orthogonal Gaussians to extract the local ordinal measures from palmprints. In [9], Guo *et al.* proposed another coding method by binarizing a palmprint image's responses to the real Gabor filters along six different orientations and they named their method as *binary orientation co-occurrence vector* (BOCV). BOCV can get a state-of-the-art verification accuracy so far. Recently, another two palmprint verification methods were proposed [10], [11] and they could achieve higher verification accuracies than BOCV; but both of them have much higher computational complexities for feature extraction and matching than BOCV.

As stated, a bit at a specific location of a code map generated by a coding scheme discussed above is typically binarized from the image's response to a filter. Consider multiple images of the same palmprint. Across all images, responses for that location will be similar, but not exactly the same. Similarly, the bit from the binary quantization could be the same across all code maps, or it may differ in some of the code maps. A bit in a palmprint's code map is consistent if it has the same value for most images of that palmprint; otherwise, it is considered as fragile. Actually, the concept of fragile bit and its effects on IrisCode have been well studied in [12]–[14]. However, to our knowledge, no investigation about fragile bits has been reported for palmprint recognition in the literature.

In this paper, we attempt to give a thorough study on the effects of fragile bits for palmprint recognition in the context of a specific coding scheme—binary orientation co-occurrence vector (BOCV) [9], which is a state-of-the-art coding method for palmprint recognition. To this end, at first, a straightforward method for identifying fragile bits for BOCV is proposed. Then, we propose to improve the verification performance of BOCV by masking out the fragile bits when calculating the Hamming distance. In addition, we believe that fragile bit patterns can also bring in some beneficial information for palmprint recognition. Thus, we propose a new metric *fragile-bit pattern distance* (FPD) to quantitatively measure the coincidence of the fragile-bit patterns in two BOCV code maps. Then, FPD is fused with the modified Hamming distance to further improve the verification accuracy of the conventional BOCV scheme. Our claims are validated by extensive experiments performed on the PolyU palmprint dataset [15]. This paper focuses only on the

Manuscript received June 11, 2012; revised July 18, 2012; accepted July 26, 2012. Date of publication August 03, 2012; date of current version August 13, 2012. This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2100219033. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiao-Ping Zhang.

L. Zhang is with the School of Software Engineering, Tongji University, Shanghai 201804, China (e-mail: cslinzhang@tongji.edu.cn).

H. Li is with the School of Software Engineering, Tongji University, Shanghai 201804, China and also with the Department of Electronic Engineering, Fudan University, Shanghai 200433, China (e-mail: hyli@tongji.edu.cn).

J. Niu is with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: jyniu@fudan.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2012.2211589

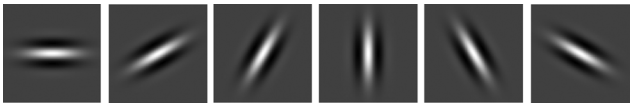


Fig. 1. Real Gabor filters along six different orientations.

feature extraction and matching stage of a palmprint recognition system. For the data preprocessing of palmprint images, such as the ROI (region of interest) extraction, please refer to [1] for details.

The remainder of this paper is organized as follows. Section II presents a method for identifying fragile bits. Section III discusses how to extend the original BOCV by making use of fragile bits information. Section IV reports the experimental results and Section V concludes the paper.

II. Fragile Bits in BOCV

As pointed out in [13], [14], bit fragility generally occurs when the inner product between a filter and a particular part of the examined image produces a value with a small magnitude. Hence, the fragility of each bit in a code map depends on a combination of the palmprint structure at that particular location, the filter adopted by the coding scheme, and the quantization method for the filter response. We put our focus on investigating the effects of fragile bits in the context of a specific coding method—BOCV [9], since it can achieve a state-of-the-art verification accuracy.

The neurophysiology-based Gabor filter proposed by Lee [16] is used in BOCV. It is defined as

$$G_j(x,y) = \frac{\omega}{\sqrt{2\pi}\kappa} e^{-(\omega^2/8\kappa^2)\left(4x'^2+y'^2\right)} \left(\cos\omega x' - e^{-(\kappa^2/2)}\right) \tag{1}$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, $\theta_j = j\pi/6$, $j = \{0,1,2,3,4,5\}$. In (1), ω is the radial frequency in radians per unit length and θ_j is the orientation of the Gabor filter. κ is defined by $\kappa = \sqrt{2\ln 2}(2^\delta + 1/2^\delta - 1)$, where δ is the half-amplitude bandwidth of the frequency response. ω can be determined by $\omega = \kappa/\sigma$, where σ is the standard deviation of the Gaussian envelop. Examples of such real Gabor filters are shown in Fig. 1 and their parameters are set as $\sigma = 8.0$, $\delta = 1.60$.

The computational architecture of BOCV is rather simple and straightforward. A BOCV code map consists of six bit-planes, each of which is binarized from the image's response to a real Gabor filter along a specific orientation. Such a process is illustrated in Fig. 2. Fig. 2(a) is a palmprint ROI image I while Fig. 2(b) are the 6 corresponding bit-planes binarized from I's responses to 6 Gabor filters with 6 different orientations.

For a filter applied to a specific location in a single image, if the response has a large magnitude, then the corresponding bit will likely be consistent. On the contrary, if the response is close to zero, the corresponding bit will be very likely to be fragile. Based on this analysis, we can use a simple method to figure out the potential fragile bits in a single BOCV code map approximately, similar as Hollingsworth *et al.* did for IrisCode [14]. Given a palmprint ROI image, suppose that we have obtained its response R_j to the j^{th} $(j = 0,1,\ldots,5)$ Gabor filter. By binarizing R_j, we can get the j^{th} BOCV bit-plane P_j. At the same time, we can sort magnitude values contained in the matrix R_j to identify p percent smallest ones. Then, we regard the corresponding bits in P_j binarized from these smallest magnitudes as fragile. Fragility mask is stored in a separate matrix f_j. Consistent bits are represented as ones while fragile bits are marked as zeros in the fragility mask. Thus, $f_j(x,y)$ indicates whether the bit $P_j(x,y)$ is consistent or not.

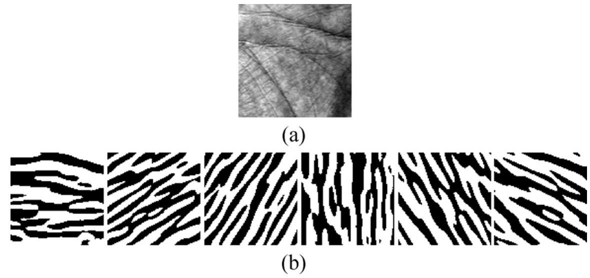


Fig. 2. (a) is a palmprint ROI image I; (b) are the 6 bit-planes binarized from I's responses to a set of Gabor filters defined by (1) with 6 different orientations.

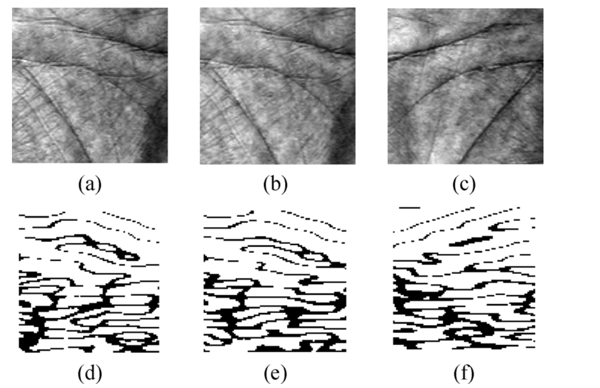


Fig. 3. (a) and (b) are the different images taken from the same palmprint while (c) is captured from a different palmprint; (d) is a fragility mask "slice" for (a); (e) is a fragility mask "slice" for (b); (f) is a fragility mask "slice" for (c). Black pixels are bits masked as fragile.

Examples of fragility masks are shown in Fig. 3. In Fig. 3, (a) and (b) are different images captured from the same palmprint while (c) is an image captured from another different palmprint. We convolve the images with G_0. Then, for each image, the locations with the smallest 25% magnitudes are masked as fragile (black pixels). Fragility masks generated from (a), (b), and (c) are shown in (d), (e), and (f). It needs to be noted that each image actually has 6 such fragility masks since BOCV uses 6 Gabor filters $\{G_j : j = 0 \sim 5\}$. Then, these 6 mask "slices" form the final fragility mask matrix.

III. Extended BOCV

In this section, two ways to improve the verification accuracy of the conventional BOCV scheme will be presented in detail.

A. Fragile Bit Masking

The original BOCV scheme adopts the normalized Hamming distance to match two BOCV code maps. In the following, we will use HD to represent the conventional Hamming distance. HD weights all bits in a code map equally. However, actually not all of the bits in a code map are equally useful. Instead, fragile bits tend to magnify an intra-class matching distance. According to this consideration, we propose to mask out fragile bits when computing the Hamming distance. With this modification, the matching distance in a comparison is based on fewer bits, but each bit used is more consistent.

The modified Hamming distance can be computed as the following. Suppose that P and Q are two BOCV code maps. Their fragility mask matrices are f and g, respectively. Then, the modified Hamming distance, denoted by HD_M, is defined as

$$HD_M = \frac{\sum_{y=1}^{rows}\sum_{x=1}^{cols}\sum_{i=1}^{6}(P_i(x,y) \otimes Q_i(x,y)) \cap f_i(x,y) \cap g_i(x,y)}{6 \cdot (f_i(x,y) \cap g_i(x,y))} \quad (2)$$

where P_i (Q_i) is the i^{th} bit-plane of P (Q), $\otimes$ represents the bitwise "exclusive OR" operation, and $\cap$ means bitwise "AND" operation.

B. Fragile-Bit Pattern Distance

From observation we find that the locations of fragile bits tend to be consistent across different code maps of the same palmprint while they quite disagree with each other in code maps from different palmprints as illustrated by examples shown in Fig. 3. This implies that further useful information could be extracted from fragile-bit patterns. To this end, we present a new metric *fragile-bit pattern distance* (*FPD*) to quantitatively measure the dissimilarity of two fragile-bit patterns.

Consider two fragile bit pattern matrices, f and g. Their FPD is defined as

$$FPD = \frac{\sum_{y=1}^{rows}\sum_{x=1}^{cols}\sum_{i=1}^{6} f_i(x,y) \otimes g_i(x,y)}{6 \cdot S} \quad (3)$$

where S is the area of the code map. In practice, taking into account the possible translations in the extracted ROI sub-image with respect to the one extracted in the enrolment, multiple matches are performed by translating one set of features in horizontal and vertical directions. And the minimum of the resulting matching distances is considered to be the final matching distance. In such cases, S is the area of the overlapping parts of the two code maps.

Even though FPD is not as powerful as HD_M, we can combine them together in order to create a better classifier. We adopt a simple weighted-average fusion rule to fuse HD_M and FPD together as

$$D = \alpha HD_M + (1-\alpha)FPD \quad (4)$$

where $0 \leqslant \alpha \leqslant 1$ is a parameter to control the contribution of HD_M in the fusion. In the following, we refer the method using D to calculate the matching distance as Extended-BOCV, or E-BOCV for short.

IV. Experimental Results and Discussions

A. Dataset, Test Protocol, and Parameter Settings

Experiments were conducted on the benchmark PolyU palmprint dataset [15], which contains 7,752 images captured from 384 different palms. In that dataset, sample images for each subject were collected in two sessions.

TABLE I
Verification Performance by Using HD and HD_M in the Context of BOCV

Model	EER
HD	0.03677%
HD$_M$	**0.03365%**

In our experiments, we took images collected in the first session as the gallery set and images collected at the second session as the probe set. Under such experimental settings, the gallery set contained 3,889 images while the probe set contained 3,863 images. To obtain statistical results, each image in the probe set was matched with all the images in the gallery set. If the two images were from the same class, the matching between them was counted as a genuine matching; otherwise it was counted as an imposter matching. The equal error rate (EER), which is the point where the false accept rate (FAR) is equal to the false reject rate (FRR), is used to evaluate the verification accuracy. Besides, by adjusting the matching threshold, a detection error tradeoff (DET) curve, which is a plot of FRR against FAR for all possible thresholds, can be created. Thus, the DET curves obtained by methods evaluated will also be provided.

Parameters involved were tuned based on a sub-dataset which contained the first 192 palmprints of the PolyU palmprint dataset and the tuning criterion was that parameter values that could lead to a lower EER would be chosen. As a result, the parameters used in this paper were set as: $\sigma = 4.6$, $\delta = 1.38$, $p = 0.08$, and $\alpha = 0.45$.

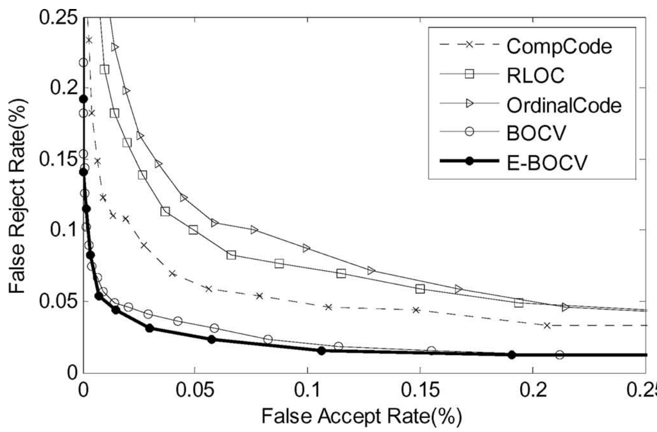
Considering the imperfectness of the ROI extraction step, we shifted the code maps vertically and horizontally in a small range when matching. The minimal distance obtained by shift matching was taken as the final matching distance. The shift range was set as $[-4, 4]$ in the following experiments.

B. Effectiveness of Fragile Bit Masking

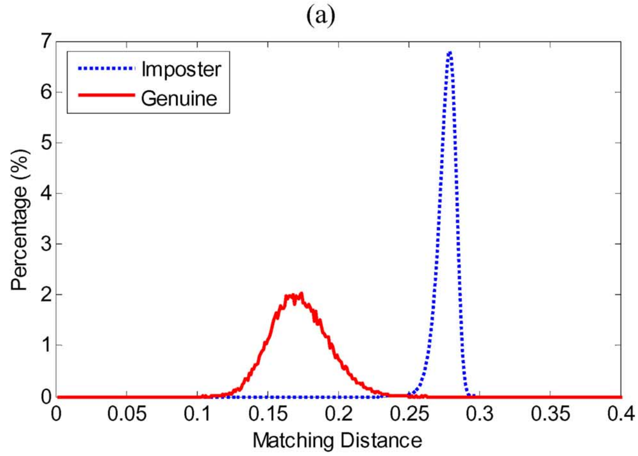
In this experiment, we compared the verification accuracies obtained by using HD and HD_M respectively in the context of BOCV. Their EERs are listed in Table I. From the results we can see that by masking out fragile bits in code maps the EER could be reduced from 0.03677% to 0.03365%. The drop of EER is 8.49% ($(0.03677-0.03365)/0.03677$). It demonstrates that the palmprint verification accuracy could be improved by masking out fragile bits.

C. Performance Evaluation of E-BOCV

Compared with the original BOCV scheme, the novelty of E-BOCV lies in the matching method it adopts. Specifically, it fuses the modified Hamming distance and the fragile-bit pattern distance to compute the dissimilarity of two code maps. In this experiment, its verification performance was evaluated and compared with the other four state-of-the-art coding based palmprint verification methods, including CompCode [6], RLOC [7], OrdinalCode [8], and BOCV [9]. The results in terms of EER are summarized in Table II. Fig. 4(a) shows the DET curves generated by the five different palmprint verification methods. Distance distributions of genuine matchings and imposter matchings obtained by the proposed E-BOCV scheme are plotted in Fig. 4(b).



(a)



(b)

Fig. 4. (a) DET curves obtained by using various palmprint verification methods; (b) distance distributions of genuine matchings and imposter matchings obtained by E-BOCV.

TABLE II
VERIFICATION PERFORMANCE OF DIFFERENT SCHEMES

	EER (%)
E-BOCV	**0.03160%**
CompCode [6]	0.05743%
RLOC [7]	0.07934%
OrdinalCode [8]	0.09156%
BOCV [9]	0.03677%

From the results listed in Table II and the DET curves shown in Fig. 4(a), we can see that E-BOCV performs the best in terms of the verification accuracy among all the state-of-the-art palmprint verification methods evaluated. With our experimental settings, the EER achieved by the proposed E-BOCV scheme was 0.0316%. Compared with the conventional BOCV method, the drop of EER is 14.06% ($(0.03677 - 0.03160)/0.03677$), which is quite significant. Therefore, the experimental results clearly corroborated our claim that by appropriately making use of fragile bits information, the performance of a palmprint coding scheme could be largely improved.

V. CONCLUSIONS

This paper is the first work discussing the fragility of bits in a palmprint code map. We investigated the effects of fragile bits in palmprint recognition in the context of a specific coding scheme, BOCV. We proposed to mask out the fragile bits when computing the Hamming distance. Furthermore, we proposed a new metric FPD to measure the dissimilarity of two fragile-bit masks. By fusing the modified Hamming distance and the FPD together, we extended the original BOCV to E-BOCV. The effectiveness of the proposed methods was corroborated by experiments conducted on the benchmark palmprint dataset.

REFERENCES

[1] D. Zhang, W. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[2] A. K. Jain and J. Feng, "Latent palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1032–1047, Jun. 2009.

[3] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1408–1418, Jul. 2009.

[4] D. Zhang, W. Zuo, and F. Yue, "A comparative study of palmprint recognition algorithms," *ACM Comput. Surv.*, vol. 44, no. 1, pp. 2:1–37, Jan. 2012.

[5] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[6] W. K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2004, pp. 520–523.

[7] W. Jia, D. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognit.*, vol. 41, no. 5, pp. 1504–1513, May 2008.

[8] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint representation for personal identification," in *Proc. IEEE Int. Conf. Computer Vis. and Pattern Recognit.*, 2005, pp. 279–284.

[9] Z. Guo, D. Zhang, L. Zhang, and W. Zuo, "Palmprint verification using binary orientation co-occurrence vector," *Pattern Recognit. Lett.*, vol. 30, no. 13, pp. 1219–1227, Oct. 2009.

[10] W. Zuo, Z. Lin, Z. Guo, and D. Zhang, "The Multiscale competitive code via sparse representation for palmprint verification," in *Proc. IEEE Int. Conf. Computer Vis. and Pattern Recognit.*, 2010, pp. 2265–2272.

[11] F. Yue, W. Zuo, and D. Zhang, "ICP registration using principal line and orientation features for palmprint alignment," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 3069–3072.

[12] R. M. Bolle, S. Pankanti, J. H. Connell, and N. Ratha, "Iris individuality: A partial iris model," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2004, pp. 927–930.

[13] K. P. Hollingsworth, K. W. Bowyer, and F. J. Flynn, "The best bits in an iris code," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 964–973, Jun. 2009.

[14] K. P. Hollingsworth, K. W. Bowyer, and F. J. Flynn, "Improved iris recognition through fusion of Hamming distance and fragile bit distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2465–2476, Dec. 2011.

[15] PolyU Palmprint Database 2006 [Online]. Available: http://www.comp.polyu.edu.hk/~biometrics

[16] T. S. Lee, "Image representation using 2D Gabor wavelet," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 957–971, Oct. 1996.