

Comparing Binary Iris Biometric Templates based on Counting Bloom Filters

C. Rathgeb and C. Busch

da/sec Biometrics and Internet Security Research Group
Hochschule Darmstadt, Darmstadt, Germany
{christian.rathgeb,christoph.busch}@h-da.de

Abstract. In this paper a binary biometric comparator based on Counting Bloom filters is introduced. Within the proposed scheme binary biometric feature vectors are analyzed and appropriate bit sequences are mapped to Counting Bloom filters. The comparison of resulting sets of Counting Bloom filters significantly improves the biometric performance of the underlying system. The proposed approach is applied to binary iris-biometric feature vectors, i.e. iris-codes, generated from different feature extractors. Experimental evaluations, which carried out on the CASIA-v3-Interval iris database, confirm the soundness of the presented comparator.

1 Introduction

Iris biometric recognition [2] is field-proven as a robust and reliable biometric technology. The iris's complex texture and its apparent stability hold tremendous promise for applying iris recognition in diverse application scenarios, such as border control or forensic investigations [12]. Daugman's algorithm [3], forms the basis of the vast majority of today's iris recognition systems, which report (true positive) identification rates above 99% and equal error rates less than 1%: (1) at enrollment an image of a subject's eye is acquired; (2) in the pre-processing step the boundary of the pupil and the outer iris are detected and the iris (in the approximated form of a ring) is "un-rolled" to obtain a normalized rectangular iris texture; (3) feature extraction is applied in order to generate a highly discriminative binary feature vector called iris-code; (4) at the time of authentication pairs of iris-codes are efficiently compared by calculating the Hamming distance between them, where template alignment is performed within a single dimension, applying a circular shift of iris-codes, to compensate against head tilts of a certain degree. While most approaches to iris recognition algorithms focus on extracting highly discriminative iris-codes, potential improvements within comparators are frequently neglected.

The contribution of this work is the proposal of a binary biometric comparator based on Counting Bloom filters (CBFs) [1,5]. In the presented scheme iris-codes are transformed to sets of CBFs which enables an enhanced biometric comparison, yielding a significant improvement in biometric performance. In

addition the generic comparator does not require a re-enrollment of registered subjects, i.e. it can be integrated to any existing iris recognition system.

This paper is organized as follows: related work is summarized in Sect. 2. The proposed comparator based on CBFs is described in detail in Sect. 3. Experiments are presented in Sect. 4 and conclusions are drawn in Sect. 5.

2 Template Comparison in Iris Recognition

Focusing on iris recognition, a binary representation of biometric features offers two major advantages:

1. Rapid authentication (even in identification mode).
2. Compact storage of biometric templates.

Comparisons between binary biometric feature vectors are commonly implemented by the simple Boolean exclusive-OR operator (XOR) applied to a pair of binary biometric feature vectors, masked (AND'ed) by both of their corresponding mask templates to prevent occlusions caused by eyelids or eyelashes from influencing comparisons. The XOR operator \oplus detects disagreement between any corresponding pair of bits, while the AND operator \cap ensures that the compared bits are both deemed to have been uncorrupted by noise. The norms ($\|\cdot\|$) of the resulting bit vector and of the AND'ed mask template are then measured in order to compute a fractional Hamming distance (HD) as a measure of the (dis-)similarity between pairs of binary feature vectors $\{\text{codeA}, \text{codeB}\}$ and the according mask bit vectors $\{\text{maskA}, \text{maskB}\}$ [3]:

$$HD = \frac{\|(\text{codeA} \oplus \text{codeB}) \cap \text{maskA} \cap \text{maskB}\|}{\|\text{maskA} \cap \text{maskB}\|}. \quad (1)$$

Note that for the dis-similarity metrics the score for a genuine comparison (i.e. both codes stemming from the same source) is expected to be low. Apart from the fractional Hamming distance several other techniques of how to compare iris-codes have been proposed. To obtain a representative user-specific iris template during enrollment Davida *et al.* [4] and Ziauddin and Dailey [13] analyze several iris-codes. Davida *et al.* propose a majority decoding where the majority of bits is assigned to according bit positions in order to reduce HD s between genuine iris-codes. Experimental results are omitted. Ziauddin and Dailey suggest to assign weights to each bit position, defining the stability of bits at according positions. Hollingsworth *et al.* [6] examined the consistency of bits in iris-codes resulting from different parts of the iris texture. The authors suggest to mask out so-called "fragile" bits for each subject, where these bits are detected from several iris-code samples. In experiments the authors achieve a significant performance gain. Obviously, applying more than one enrollment sample yields better recognition performance, however, commercial applications usually require single sample enrollment as the operational constraints can not tolerate an extended capture process duration. Rathgeb *et al.* [10,11] have demonstrated that incorporating preliminary comparison scores, which are obtained during the

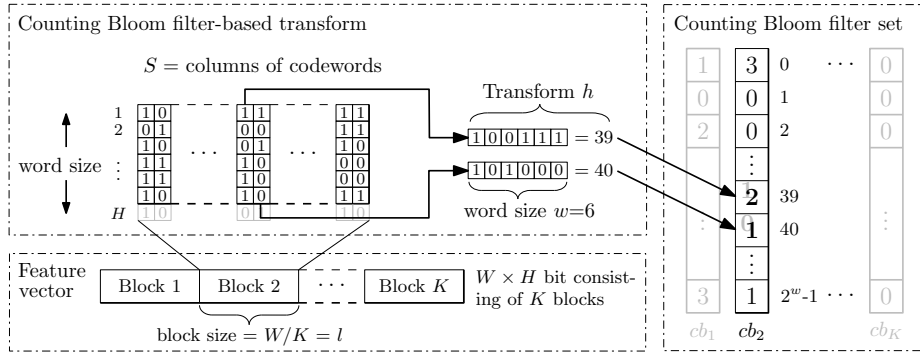


Fig. 1. Proposed Counting Bloom filter-based transform: highlighted codewords increment in cb_2 the element at index 39 and 40.

alignment process, significantly increases biometric performance. HD scores are expected to decrease towards an optimal alignment, i.e. the distance between the lowest and highest score as well as the overall distribution yielded by scores at different shifting positions, indicates (non-)genuine comparisons. Typically, minor improvements do not lead to significant performance gain with respect to accuracy. On the other hand, more complex comparison techniques do not provide a rapid comparison of biometric templates, yielding a trade-off between computational effort and recognition accuracy.

3 Counting Bloom Filter-based Comparator

Basically, a Bloom filter b is a bit array of length n , where initially all bits are set to 0 [1]. In order to represent a set S a Bloom filter traditionally utilizes k independent hash functions h_1, h_2, \dots, h_k with range $[0, n - 1]$. For each element $x \in S$, bits at positions $h_i(x)$ of Bloom filter b are set to 1, for $1 \leq i \leq k$. To test if an element y is in S , it has to be checked whether all position of $h_i(y)$ in b are set to 1. If this is the case, it is assumed that y is in S with a certain probability of false positive. If not, clearly y is not a member of S , hence, traditional Bloom filters are suitable for any application where a distinct probability of false positive is acceptable. In a Counting Bloom filter cb , which has first been introduced by Fan et al. [5], the array positions are extended from being a single bit to being an integer counter.

In the following subsections, the CBF-based transform, which is depicted in Fig. 1, and the corresponding comparison technique are described in detail.

3.1 Counting Bloom Filter-based Transform

In the proposed system CBFs are utilized in order to achieve an alignment-free representation (to a certain degree) of iris-codes. For this purpose the original concept of CBFs is adapted in two ways:

Algorithm 1 Construction of CBF-based template.

```
for  $j = 0 \rightarrow K - 1$  do ▷ process each block of the feature vector
  for  $i = j * l \rightarrow j * (l + 1)$  do ▷ process each codeword within a block
     $cb_j[h(x_i)] \leftarrow cb_j[h(x_i)] + 1$  ▷ increment the CBF at the according position
  end for
end for
```

1. A single trivial transform h is utilized instead of numerous hash function.
2. A fixed number of exactly l elements are inserted into an according CBF.

Generic iris recognition systems [2] extract binary feature vectors based on a row-wise analysis of normalized iris textures, i.e. iris-codes typically represent two-dimensional binary feature vectors of width W and height H (see Fig. 2 (e)-(f)). In the proposed scheme $W \times H$ iris-codes are divided into K blocks of equal size, where each column consists of $w \leq H$ bits. In case $w < H$, columns consist of the w upper most bits, i.e. features originating from outer iris bands, which are expected to contain less discriminative information, are ignored and not represented in the CBF. Subsequently, the entire sequence of columns of each block is successively transformed to according locations within CBFs, that is, a total number of K separate Bloom filters of length $n = 2^w$ form the template of size $K \cdot 2^w$. The transform is implemented by mapping each column in the iris-code to the index of its decimal value, which is shown for two different codewords (=columns) as part of Fig. reffig:system, for each column $x \in \{0, 1\}^w$, the mapping is defined as,

$$h(x) = \sum_{j=0}^{w-1} x_j \cdot 2^j. \quad (2)$$

The entire process of constructing a set of CBFs which represents a distinct iris-code is described in Algorithm 1. The representation is alignment-free, i.e. generated templates (=sets of CBFs) do not need to be aligned at the time of comparison. Equal columns within certain blocks (=codewords) increment identical indexes within CBFs, i.e. self-propagating errors caused by an inappropriate alignment of iris-codes are eliminated (radial neighborhoods persist).

3.2 Comparison in transformed Domain

The dissimilarity DS between two CBFs cb and cb' of length $n, n = 2^w$, is defined as the sum of difference at each index of both CBFs,

$$DS(cb, cb') = \sum_{j=1}^n |cb_j - cb'_j|/2l, \quad (3)$$

Obviously, DS requires more computational effort compared to HD , however, DS does not have to be computed at numerous shifting positions. In order to incorporate masking bits obtained at the time of pre-processing, columns of iris-codes which are mostly affected by occlusions must not be mapped to Bloom filters, i.e. a separate storage of bit masks is not required.

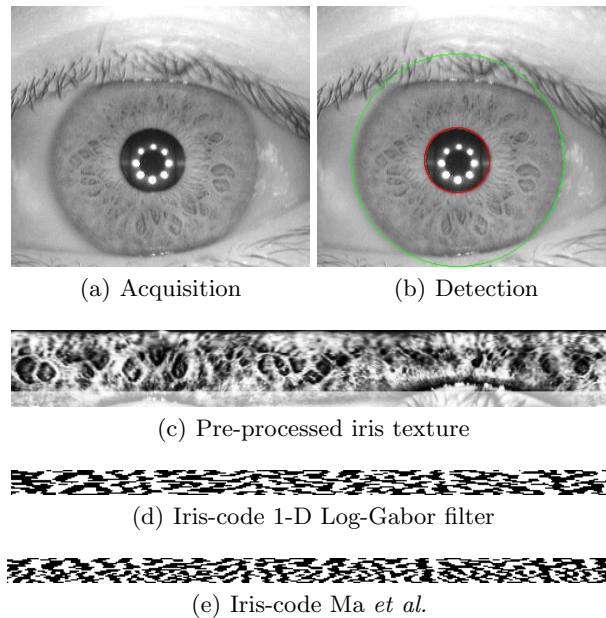


Fig. 2. Iris processing chain: applied pre-processing and feature extraction algorithms.

4 Experiments

Performance is estimated in terms of false non-match rate (FNMR) at a targeted false match rate (FMR) and equal error rate (EER). In accordance to the International Standard ISO/IEC IS 19795-1 [7] the FNMR of a biometric system defines the proportion of genuine attempt samples falsely declared not to match the template of the same characteristic from the same user supplying the sample. By analogy, the FMR defines the proportion of zero-effort impostor attempt samples falsely declared to match the compared non-self template. As score distributions overlap EERs are obtained, i.e. the system error rate where $\text{FNMR} = \text{FMR}$.

4.1 Experimental Setup

Experiments are carried out using the CASIA-v3-Interval iris database¹. At pre-processing the iris of a given sample image is detected, un-wrapped to an enhanced rectangular texture of 512×64 pixel, shown in Fig. 2 (a)-(d).

In the feature extraction stage custom implementations² of two different iris recognition algorithms are employed where normalized iris textures are divided

¹ The Center of Biometrics and Security Research,
<http://www.idealtest.org>

² USIT – University of Salzburg Iris Toolkit v1.0,
<http://www.wavelab.at/sources/>

Table 1. Original performance (in %) for both feature extractors (*HD* comparator).

Algorithm	1-FNMR @ FMR=0.01	EER
1-D Log Gabor	95.03	1.58
Ma <i>et al.</i>	96.16	1.19

Table 2. 1-FNMRs @FMR=0.01 (in %) for different configurations of the comparator.

Algorithm	Word size w (bits)	Block size l (bits)				
		2^5	2^6	2^7	2^8	2^9
1D Log Gabor	10	95.75	94.32	88.43	66.24	31.81
	9	94.98	94.27	89.36	64.45	–
	8	93.65	93.91	87.97	–	–
Ma <i>et al.</i>	10	98.15	96.11	93.40	82.65	60.71
	9	97.80	94.88	91.30	76.21	–
	8	97.08	93.40	87.92	–	–

into stripes to obtain 10 one-dimensional signals, each one averaged from the pixels of 5 adjacent rows (the upper 512×50 rows are analyzed). The first feature extraction method follows an implementation by Masek [9] in which filters obtained from a Log-Gabor function are applied. A row-wise convolution with a complex Log-Gabor filter is performed on the texture pixels and the phase angles of resulting complex values are discretized into 2 bits generating a binary code of $512 \times 20 = 10240$ bit. The second feature extraction algorithm was proposed by Ma *et al.* [8]. Within this algorithm a dyadic wavelet transform is performed on 10 signals obtained from the according texture stripes. For two selected subbands minima and maxima above an adequate threshold are located, and a bit-code of $512 \times 20 = 10240$ bits is extracted. Sample iris-codes generated by both feature extraction methods are shown in Fig. 2 (e)-(f). iris-code are divided into upper and lower 512×10 halves as these represent real and complex values or minima and maxima extracted from different subbands, respectively.

4.2 Performance Evaluation

The biometric performance of the original systems, in which *HD*-based iris-code comparisons are performed at ± 8 circular bit shifts, are shown in Table 1. The corresponding receiver operation characteristic (ROC) curves are plotted in Fig. 3 (a). For both feature extraction techniques practical performance rates are achieved, yielding EERs of 1.58% and 1.19%, respectively. With respect to the proposed CBF-based comparator, Table 2 and Table 3 summarize obtained 1-FNMRs at target FMRs of 0.01% and EERs for different word sizes w and block sizes l for both feature extraction algorithms. As can be seen, a choice of large block sizes implies a greater loss of local information (original positions of

Table 3. EERs (in %) for different configurations of the proposed comparator.

Algorithm	Word size w (bits)	Block size l (bits)				
		2^5	2^6	2^7	2^8	2^9
1D Log Gabor	10	1.21	1.75	2.49	4.54	7.87
	9	1.34	1.77	3.02	4.74	–
	8	1.42	1.93	3.17	–	–
Ma <i>et al.</i>	10	0.88	1.56	2.54	4.10	6.90
	9	1.04	1.61	2.70	4.62	–
	8	1.09	1.67	3.22	–	–

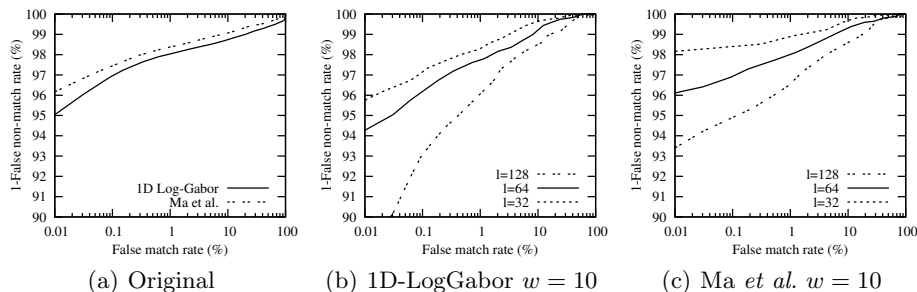


Fig. 3. ROC curves for (a) the original *HD*-based comparator and the proposed algorithm for (b) the 1D Log-Gabor feature extractor and (c) the algorithm of Ma *et al.* for different settings of block sizes and a word size of $w = 10$.

codewords) and causes a drastic decrease in biometric performance. From the obtained results it is clear that rotations of ± 8 bits, which significantly affect original systems, are compensated. For both feature extraction algorithms performance is gained for different configurations, achieving best results at word size of $w = 10$ and a block size of $l = 32$, obtaining EERs of 1.21% and 0.88%, respectively. The according ROC curves for a word size of $w = 10$ are depicted in Fig. 3 (b)-(c). Significant improvement is obtained compared to the original system, while the proposed scheme does not require re-enrollment or any adaption of the original iris-codes. CBFs can be stored in addition to iris-code records or efficiently calculated at the time of comparison.

5 Conclusions

In this work an advanced binary biometric comparator based on counting Bloom filters has been introduced. Compared to a conventional, *HD*-based comparison, within the proposed approach iris-codes are transformed to sets of CBFs, prior to comparison. Additional computational effort is limited since the CBF-based representation enables an alignment-free comparison. The system is evaluated

on a publicly available dataset where it gains biometric performance for different feature extraction techniques, confirming the soundness of the presented approach.

Acknowledgment

This work has been partially funded by the European FP7 FIDELITY project (SEC-2011-284862) and the Center of Applied Security Research Darmstadt (CASED).

References

1. B. Bloom. Space/time tradeoffs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
2. K. Bowyer, K. Hollingsworth, and P. Flynn. Image understanding for iris biometrics: A survey. *Computer Vision and Image Understanding*, 110(2):281–307, 2007.
3. J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.
4. G. Davida, Y. Frankel, and B. Matt. On enabling secure applications through off-line biometric identification. In *Proc. IEEE Symp. on Security and Privacy*, pages 148–157. IEEE, 1998.
5. L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, June 2000.
6. K. P. Hollingsworth, K. W. Bowyer, and P. J. Flynn. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):964–973, 2009.
7. ISO/IEC TC JTC1 SC37 Biometrics. *ISO/IEC 19795-1:2006. Information Technology – Biometric Performance Testing and Reporting – Part 1: Principles and Framework*. International Organization for Standardization and International Electrotechnical Committee, Mar. 2006.
8. L. Ma, T. Tan, Y. Wang, and D. Zhang. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image Processing*, 13(6):739–750, 2004.
9. L. Masek. Recognition of human iris patterns for biometric identification. Master’s thesis, University of Western Australia, 2003.
10. C. Rathgeb, A. Uhl, and P. Wild. Shifting score fusion: On exploiting shifting variation in iris recognition. In *Proc. 26th ACM Symposium On Applied Computing*, pages 1–5. ACM, 2011.
11. C. Rathgeb, A. Uhl, and P. Wild. Iris-biometric comparators: Exploiting comparison scores towards an optimal alignment under gaussian assumption. In *Proc. 5th Int’l Conf. on Biometrics*, pages 1–6. IEEE, 2012.
12. C. Rathgeb, A. Uhl, and P. Wild. *Iris Biometrics: From Segmentation to Template Security*. Number 59 in *Advances in Information Security*. Springer, 2012.
13. S. Ziauddin and M. Dailey. Iris recognition performance enhancement using weighted majority voting. In *Proc. 15th Int’l Conf. on Image Processing*, pages 277–280. IEEE, 2008.