Iris-Biometric Hash Generation for Biometric Database Indexing

Christian Rathgeb and Andreas Uhl
Department of Computer Sciences, University of Salzburg, 5020 Salzburg, Austria
{crathgeb, uhl}@cosy.sbg.ac.at

Abstract

Performing identification on large-scale biometric databases requires an exhaustive linear search. Since biometric data does not have any natural sorting order, indexing databases, in order to minimize the response time of the system, represents a great challenge.

In this work we propose a biometric hash generation technique for the purpose of biometric database indexing, applied to iris biometrics. Experimental results demonstrate that the presented approach highly accelerates biometric identification.

1 Introduction

In biometric authentication two different modes, in which a system can be operated, are distinguished – verification and identification [4]. While verification involves a single $1:1$ comparison, identification involves a $1:n$ comparison, where n is the number of persons registered with the system. For a large-scale database, where a single identification may involve millions of matching procedures computational limits are reached relatively fast [4]. To reduce the computational effort, serial combinations of algorithms [6] have been proposed. In a serial combination approach an efficient pre-selection algorithm is applied to the whole database and a more complex, but more accurate, algorithm is applied to a small subset of selected users. Regardless of the efficiency of the pre-selection algorithm, for a large-scale database computational effort will only be reduced to a certain extent. Focusing on iris biometrics best experimental results were presented by Gentile *et al.* [1] who reduced the number of required bit-comparisons of iris-codes to less than 10%. Biometric database indexing [7], which requires a classification of biometric data [3, 9], turns out to be a very recent field of research. The aim of indexing biometric databases is to perform a coarse level classification of templates and accurate matching afterwards. Approaches to database indexing require complex data structures where the insertion of new users may result in a re-classification of the whole dataset [7]. Until now the best results were obtained by Mhatre *et al.* [7] which reduced the search space to 5% of the applied dataset for a multimodal system based on face and signature features. With respect to iris-based indexing techniques Jayaraman *et al.* [5] reduced the search space to >40% and Mukherjee and Ross [8] to 30% for a hit rate of 84%, respectively.

In this work we utilize biometric hash generation (HG) [10] for database indexing which has, to our knowledge, not been investigated until now. Based on iris recognition we propose a generic HG scheme. Biometric data is hashed such that the calculated hash points to a location in the database where the according first rank match is most likely to be found. Hence, a biometric HG is applied in order to preselect an according subset of all users.This paper is organized as follows: In Sect. 2 we describe our proposed system in detail. Experiments are presented in Sect. 3. Sect. 4 concludes this paper.

2 System Architecture

The basic idea of our approach is to apply biometric hashing for database indexing to locate biometric templates in the database at a coarse level. Hence, in contrast to conventional approaches no complex sorting of templates is required since we do not sort biometric templates but use low-dimensional hashes which are directly generated out of biometric data.

Several problems arise with the introduction of biometric hashes to database indexing. Firstly, if it would be possible to reliably generate collision free hashes independent of the number of registered persons these hashes could be used to reliably identify persons. In other words, if biometric hashing would work perfectly the use of any underlying biometric algorithm would become meaningless. Yet, a perfect system is not realistic if only a fraction of bits are used to authenticate a person. Since the generation of a template can be seen

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.698

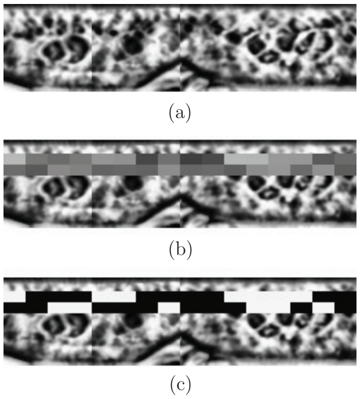

Figure 1. 32-bit HG (16×8 pixel blocks, k=128): (a) preproc. iris texture (b) mean values for pixel blocks (c) binary hash.

as HG per se, we want to restrict to determining the coarse location of templates in the database.

The proposed system is different from any generic serial combination of recognition algorithms since the pre-selection algorithm is not applied to all registered users. Serial combinations of single and multi modal biometrics may use a fast algorithm for the pre-selection and apply accurate matching to a selected subset. However, the matching procedure of the first algorithm has to be applied for each registered user. If a coarse pre-classification of biometric templates is applied accurate matching will only be performed within one class. However, existing approaches to pre-classification extract only a small number of classes (e.g. 5 classes in [9]). Thus, a considerable number of matches have to be performed within the according classes. In the proposed system the underling biometric algorithm is fed with templates which are most likely to match with the given sample. Hence, only a small number of matchings have to be performed.

2.1 Biometric Hash Generation

In our approach the top and the bottom quarter (315^o to 45^o and 135^o to 225^o) of the iris ring are discarded during preprocessing since these parts are often affected by eyelids or eyelashes. Remaining parts of the preprocessed texture are employed to extract a hash, suitable for database indexing. Calculated hashes have to fulfil several requiremts such as collision freeness as well as efficient computation. Since stable HG is necessary to overcome biometric variance, processing the most reliable parts of an iris texture is desirable. We analyze $x \times y$ pixel blocks located in the upper half of the iris texture since it was found that the inner bands of the iris are more consistent than the outer bands [2]. To overcome variance, quantization is introduced by calculating mean grayscale values for each processed $x \times y$ pixel block. For a number of n pixel blocks a set M of mean values, defined by $M = \{m_1, m_2, ..., m_n\}$, is extracted. Subsequently, each mean value m_i is compared against a predefined threshold k to generate a binary hash $B = \{b_1, b_2, ..., b_n\}$ such that,

$$b_i = \begin{cases} 1, & \text{if } m_i \geq k, \\ 0, & \text{if } m_i < k. \end{cases} \forall i = 1...n \qquad (1)$$

Fig. 1 shows the process of HG which is computational efficient and offers several advantages: the proposed HG extracts similar hashes for similar biometric data, fulfilling the requirement of a hash used for coarse level database indexing. Hashes are scalable, with respect to length, by varying block dimensions and the number of applied blocks. Since hash values are used as database indices scalability turns out to be important.

2.2 DB-Indexing and Registration

HG is applied during template generation representing a single-sensor scenario. For each user a biometric template as well as a hash are generated. Hashes are arranged in a way such that if a similar hash is obtained from another image, the according template is found quickly. For this purpose we construct a Karnaugh map (KM) [11] for n-bit hashes a priori. The time required to construct the map does not affect identification performance since it is calculated at system set up. At registration, a pointer to the generated biometric template is stored at the according node of the KM. The according node is equal to the extracted hash code. If equal hashes are calculated for different users a linked list of pointers is stored at the according node. Fig. 2 shows a KM for 4-bit codewords. One property of a KM is that any path through adjacent nodes results in a Gray code. Thus, nodes which can be reached by short paths expose high similarity applying the Hamming distance. Finding the according node of the KM which corresponds to the extracted hash can be done in logarithmic time (KMs are stored in a sorted structure). Applying binary search the registration of a user requires logarithmic effort with respect to the dimension of the KM [11].

2.3 DB-Search and Identification

At identification, biometric data is acquired, preprocessing is applied and features and a hash are extracted. One important restriction to the HG is that it has to operate as pure key generation scheme [10] since no additional user-specific helper data is available. Then the extracted hash is mapped onto the stored KM. If the according node points at one or more biometric templates,

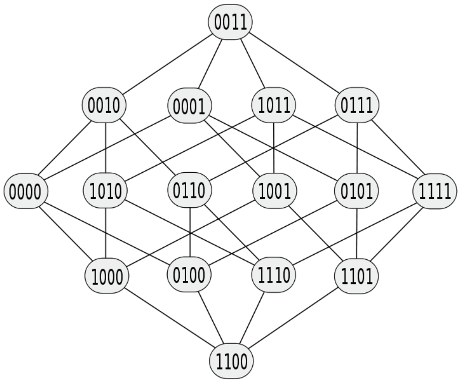

Figure 2. 4-bit KM: each node is linked with each other node which has a Hamming distance of 1 to this node.

the extracted iris-code is matched against these. This procedure is repeated recursively for neighboring nodes until identification is yielded or a certain search depth is reached. If there is no pointer present or identification fails all adjacent nodes are considered.

2.4 Computational Performance

Assume that the matching between two templates takes time T_M. A threshold-based identification of one person out of N takes an average time $\overline{T}_L$, based on linear search, of,

$$\overline{T}_L = \frac{N}{2} \cdot T_M \tag{2}$$

If a serial combination of algorithms is used where the matching procedure of a fast pre-selection algorithm takes time t_S the average time $\overline{T}_S$, based on pre-selection, to identify a person takes,

$$\overline{T}_S = N \cdot t_S + n_S \cdot T_M \tag{3}$$

where n_S is the number of person which are considered for the more complex algorithm. The pre-selection algorithm returns n_S identities which are considered for a more accurate matching ($n_S \ll N, t_S < T_M$). If,

$$\left(\frac{N}{2} - n_S\right) \cdot T_M > N \cdot t_S \tag{4}$$

computational effort is reduced, regardless of any performance decrease in terms of recognition rates. In indexed biometric databases the theoretical average time $\overline{T}_I$ to authenticate a person takes,

$$\overline{T}_I = n_I \cdot T_M + \delta_I \tag{5}$$

where δ_I is the time required to detect the according position in the applied data structure and n_I is the number of matching procedures which have to be performed until the correct identity is detected. Here n_I can not be assumed to be an even smaller number ($n_I < n_S$). Depending on the applied data structures n_I is expected to be a logarithm of N. Additionally, the insertion and deletion of users implies high effort. The application of a pre-classification of stored biometric templates will result in similar effort where n_I is even greater.

The average time $\overline{T}_H$ to identify a person in our scheme, based on biometric hashing, takes,

$$\overline{T}_H = n_H \cdot (t_H + T_M) + \delta_H \tag{6}$$

where n_H is the number of matching procedures which are necessary to detect the correct identity and t_H is the time for comparing two hashes. δ_H defines additional time required to detect correct nodes in the KM. If n_H can be held low, computational effort is reduced. Compared to a serial combination of algorithms, for a small number of n_H and δ_H,

$$\left(\frac{N}{2} - n_H\right) \cdot T_M > n_H \cdot t_H + \delta_H \tag{7}$$

will be true in most cases. As a consequence, the system is able to outperform serial combinations of algorithms if n_H is small enough. Compared to indexed biometric databases the scheme shows worse performance at first sight. However, the search time δ_I trough complex data structures in which templates are arranged yields several decision requiring reasonable effort. Thus, the computational performance of the proposed system is expected to outperform existing approaches to biometric database indexing as well. Experimental results support our claim. Additionally, enrollment only involves the insertion of a single pointer into a KM, which requires logarithmic effort.

3 Experimental Results

Experiments are carried out using the CASIAv3-Interval iris database, which comprises iris images of 250 persons ($N = 250$) where iris textures of 256×64 pixels are extracted during preprocessing. In our experiments we will first focus on the exclusive performance of the hashing algorithm. Then we will apply our system to an existing iris recognition algorithm to evaluate performance gain with respect to time consumption.

To evaluate the exclusive performance of our hashing algorithm we assume that the underlying iris recognition algorithm works perfectly. That is, if matching is performed between two templates of one person successfull identification is yielded. We are aware that this is an unrealistic assumption. Nevertheless, this assumptions clearly reveals the actual performance gain. Performance depends on the average number of matchings

Block Dim.	Hash Length	Avg. Matchings
8×8	32 bit	13.211
12×8	32 bit	10.555
16×8	32 bit	7.695
16×12	32 bit	9.671
32×4	32 bit	9.344
16×8	24 bit	11.912
32×8	24 bit	13.736
16×8	16 bit	15.697

Table 1. Average matchings for different block dimensions and hash lengths.

(n_H) required to identify a single person. Firstly several parameters have to be set up, including the applied pixel block dimension, the threshold k and the number of pixel blocks (=hash length). For several parameters the average number of matchings, which have to be performed until successful authentication is achieved, are summarized in Tab. 1 ($k = 128$). Based on applied block dimensions an appropriate number of pixel block rows are chosen to achieve the according hash length. Best result were achieved for 16×8 pixel blocks and a hash length of 32 bit. Depending on the applied pixel block dimension all hashes turned out to be collision free. Thus, no longer hashes are required.

The probability of successful identification (hit rate), according to the number of matchings which have to be performed, for the best parameters are plotted in Fig. 3. Applying the proposed algorithm for a block dimension of 16×8 and 32 bit hashes the average number of matchings which yield successful identification is 7.695. Hence, our system highly outperforms generic identification systems which perform linear matching. For large database sizes our system theoretically outperforms a serial combination of algorithms as well since the matching of hashes need not be performed for the entire dataset. Thus, the proposed system is expected to perform much faster than a serial combination. On average only $7.695/250 \simeq 3\%$ ($N = 250$) of the templates stored in the database are processed! Compared to existing approaches to iris-based serial combinations [1] of algorithms and iris-based database indexing [5, 8] obtained results are promising.

To calculate performance with respect to time consumption we apply our approach to our implementation of the algorithm of Masek[1]. Applying identification based on linear search takes 460.0198 seconds for the whole database whereas our approach yields a time consumption of 47.9807 seconds. Compared to previous results performance decreases, since the search in the KM as well as hash matching requires extra time. Still computational overhead is low and effort is reduced by 89.57 % maintaining recognition rates.

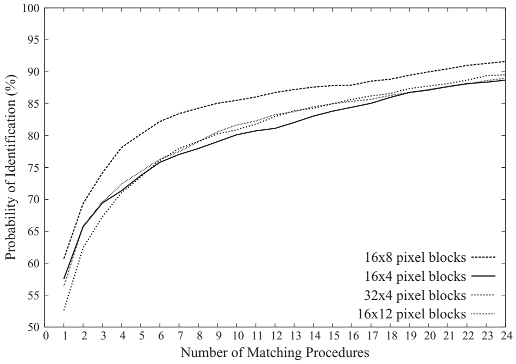

Figure 3. 32-bit HG for the most appropriate pixel block dimensions (t=128).

4 Conclusion

We presented an efficient approach to biometric database indexing based on iris. Applying our system in identification mode only $\sim$3% of the whole database have to be processed on average and a performance gain of 89.57% with respect to time consumption is obtained.

References

[1] J. Gentile, N. Ratha, and J. Connell. An efficient, two-stage iris recognition system. In *International Conference on Biometrics: Theory, Applications, and Systems (BTAS '09)*, pages 1–5, 2009.

[2] 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.

[3] A. Jain and S. Pankanti. Fingerprint classification and matching. *Handbook for Image and Video Processing*, 2000.

[4] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14:4–20, 2004.

[5] U. Jayaraman, S. Prakash, Devdatt, and P. Gupta. An indexing technique for biometric database. *International Conference on Wavelet Analysis and Pattern Recognition. ICWAPR '08.*, 2:758–763, 2008.

[6] J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[7] A. J. Mhatre, S. Palla, S. Chikkerur, and V. Govindaraju. Efficient search and retrieval in biometric databases. *Proceedings of the SPIE Defense and Security Symposium*, 5779:265–273, 2005.

[8] R. Mukherjee and A. Ross. Indexing iris images. In *International Conference on Pattern Recognition (ICPR 08)*, pages 1–4, 2008.

[9] N. Ratha, K. Karu, S. Chen, and A. Jain. A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), 1996.

[10] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[11] S. P. Vingron. Karnaugh maps. *Switching Theory: Insight Through Predicate Logic*, pages 57–76, 2004. Springer-Verlag.

[1] L. Masek, Recognition of Human Iris Patterns for Biometric Identification, Master's thesis, University of Western Australia, 2003