Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

A Novel Coding Scheme for Indexing Fingerprint Patterns*

Aglika Gyaourova and Arun Ross

West Virginia University, Morgantown WV 26506, USA
agyaourova@mix.wvu.edu, arun.ross@mail.wvu.edu

Abstract. Indexing is the process of assigning a numerical value to a database entry in order to facilitate its rapid retrieval. Indexing a fingerprint database can reduce the search space and improve the response time of an identification system. We discuss a novel method for generating index codes for fingerprint images by using a small set of pre-determined *reference* fingerprints. In the proposed method, the match scores generated by comparing an input fingerprint with the reference fingerprints are subjected to a discretization function, which converts them into an index code. A search mechanism based on the Hamming distance identifies those index codes in the database that are similar to the code of the input image. The proposed technique has several advantages: it obviates the need to extract complex features from the fingerprint image; it utilizes the matcher that is already associated with a particular application; and it can be used to index any biometric database irrespective of the trait or matcher being used. Experimental results on two fingerprint databases (NIST-4 and WVU) indicate that the proposed encoding scheme generates index codes that are well-scattered thereby allowing noisy query images to be indexed correctly.

1 Introduction

In a fingerprint identification system, the goal is to assign an input (or query) fingerprint to one of several identities present in the database. Every identity is defined by a fingerprint image and the identification operation entails the comparison of the query image with the images residing in the database. Filtering is the process of reducing the number of fingerprints to be retrieved from a database for comparison during an identification operation. Two important considerations make filtering a desirable part of large-scale fingerprint identification systems: (a) invoking a matcher on the entire database can be computationally expensive, and (b) the number of false positives grows exponentially with the size of the database [7][12]. Filtering can be accomplished by using two distinct schemes: classification or indexing. A classification scheme partitions the database of fingerprints into several classes. The query image is compared only against those images in the database belonging to the *same* class as itself. Indexing schemes,

* This work was supported by US NSF CAREER grant number IIS 0642554.

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 765–774, 2008.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

on the other hand, assign an index value to every fingerprint and the query image is compared only against those images in the database having a *comparable* index value.

While the Henry fingerprint classification system[1] [1] has been traditionally used to perform filtering, fingerprint researchers have also developed alternate methods to filter fingerprints [11]. A review of the literature indicates that these techniques either utilize image features implicit in a fingerprint [2,3,5,6,9] or employ a fingerprint matcher to describe the relationship between multiple fingerprints based on pairwise match scores [10]. While the former approach has been widely researched, the latter has received very little attention in the literature primarily due to the need for generating and processing score matrices. For example, consider the work in [10] where the query image is sequentially matched against fingerprints in the database. At each step in the sequence, a decision is made regarding the choice of the next fingerprint to be retrieved. This decision is based on finding correlations in the score matrix containing the pairwise match scores of all fingerprints in the database. While this method obviates the need to perform additional image processing to compute index values, storing the matrix of match scores for a database containing millions of fingerprints can be prohibitive.

In this work, we utilize match scores to index and retrieve fingerprint images. However, the proposed technique does not use a score matrix in order to guide its search. Rather, the method relies on comparing a fingerprint with a *small* set of reference images and using the resulting match scores to derive an index code. The proposed method can be applied to any biometric database irrespective of the biometric trait or matcher being used.[2] Furthermore, it generates a compact code based on the evidence of a single impression of a finger. Thus, the proposed method has modest storage requirements.

The rest of the paper is organized as follows. In section 2 we discuss the three separate stages of the proposed method: (a) selecting reference images, (b) creating index codes for the images in the database, and (c) retrieving pertinent fingerprints from the database based on a query image. Section 3 describes the experimental setup using ternary-valued index codes. Experimental results on two databases are presented in section 4. Section 5 summarizes the paper and describes ways to further improve the proposed method.

2 Indexing Methodology

The crux of the proposed method relies on the use of a small set of reference fingerprint images, $\boldsymbol{R} = \{r_1, r_2, ..., r_n\}$, that have good representative and discriminative power. The representative power ensures that there are a sufficient *number* of reference fingerprints, while the discriminative power ensures that

[1] The disadvantages of the Henry classification system to filter one-print databases (as opposed to ten-print databases) have been discussed in [11].

[2] Indeed, it may be applied to any pattern retrieval problem provided an appropriate pattern matcher is available.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

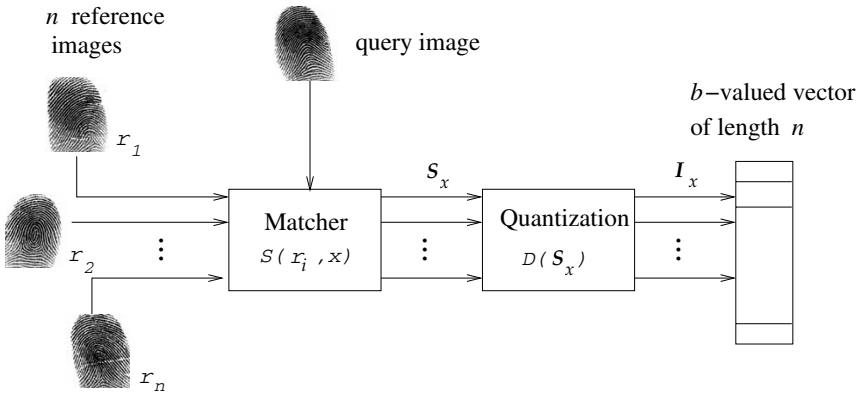

Fig. 1. Generating an index code using reference fingerprints

these prints exhibit sufficient *variability* amongst themselves. Rather than relying on implicit image features to select these reference images, the match scores generated by a fingerprint matcher are used.

The index of an arbitrary fingerprint image x is constructed in two steps. First, a set of match scores is computed by comparing x against the reference fingerprints, $r_1, r_2, ..., r_n$, in a fixed order. The result is the set $\boldsymbol{\mathcal{S}}_x = \{\mathcal{S}(r_1, x), ..., \mathcal{S}(r_n, x)\}$, where $\mathcal{S}()$ is the matching function. Next, a discretization function $\mathcal{D}_b : \mathbb{R}^n \rightarrow \mathbb{Z}_b^n$ is used to transform $\boldsymbol{\mathcal{S}}_x$ to an index code, i.e., $\boldsymbol{\mathcal{I}}_x = \mathcal{D}_b(\boldsymbol{\mathcal{S}}_x)$. Here, b is the number of distinct values that are possible for each element in the code. For example, $b = 2$ would imply that $\boldsymbol{\mathcal{I}}_x$ is a binary vector (see Figure 1).

During enrollment, the fingerprint image associated with an identity is subjected to the procedure above in order to generate its *model index code*. The model index code is stored in the fingerprint database along with the identity of the associated print. During identification, when a query image is presented to the system, a Hamming distance-based search mechanism is invoked to retrieve only those fingerprints in the database whose model index codes are probable candidates for a match.

Thus, the architecture of the proposed scheme is defined by the ordered set of reference fingerprint images ($\boldsymbol{R}$), the discretization function ($\mathcal{D}_b$), and the search mechanism used to identify potential model index codes. Figure 2 shows the framework and the information flow in the proposed indexing method.

2.1 Selecting Reference Images

Selection of the set of reference images is an important component of the proposed scheme. Successful indexing is possible only when the model index codes are well distributed over the index space. Therefore, the range of match scores obtained when comparing a reference image against different fingerprints in the database should be high. In the current implementation, the reference images are

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

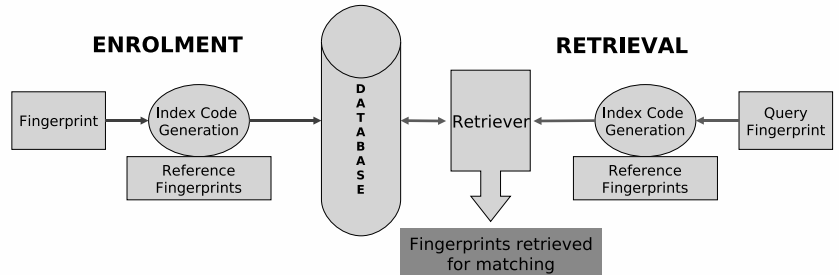

Fig. 2. Framework of the proposed method

selected from the fingerprint database itself (although they could be some type of digital artifacts as well, such as synthetic fingerprints). Fingerprint images whose impostor match scores exhibit a large variance are selected as reference fingerprints. While the entire database of prints may be viewed as a candidate pool for selecting reference images, practical considerations dictate the use of a small random subset of prints for this purpose.

Algorithm for selecting n reference fingerprints

Let $\boldsymbol{\mathcal{F}} = \{f_1, f_2, ..., f_P\}$ be the candidate pool of reference images, $\boldsymbol{\mathcal{D}} = \{d_1, d_2, ..., d_L\}$ be an "impostor" database of fingerprints, and $S()$ be the fingerprint matcher.

1. For each f_i compute $v_i = \text{Variance}\{\mathcal{S}(f_i, d_j)\}_{j=1}^{L}$.
2. Sort v_i in descending order.
3. Use the fingerprints corresponding to the top n v_i values as reference fingerprints.

There is a possibility of two reference images having very similar impostor match scores when compared against other images. Retaining both these reference images will not enhance the uniqueness of the index codes. Therefore, an additional constraint may be needed, which ensures that every pair of reference images are sufficiently dissimilar.

2.2 Creating the Index Code

The index code for an arbitrary image x is computed as follows. First, the vector $\boldsymbol{\mathcal{S}}_x$ is computed as $\boldsymbol{\mathcal{S}}_x = \{s_1^x, s_2^x, ..., s_n^x\}$, where $s_i^x = \mathcal{S}(r_i, x)$. Next, the index $\boldsymbol{\mathcal{I}}_x = \{i_1^x, i_2^x, ..., i_n^x\}$ of x is generated by using a discretization function $\mathcal{D}_b(\boldsymbol{\mathcal{S}}_x)$. In its simplest form, the discretization function applies thresholds on the match scores to convert them to a discrete domain. The choice of b, denoting the maximum number of output values of the discretization function, depends on the characteristics of the match scores generated by the matcher. For example, if

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

the scores generated when comparing an arbitrary image with multiple impressions of a fixed impostor finger are stable, than b can be as large as the number of possible match score values. On the other hand, when these scores exhibit large variation, then a small value of b would be more appropriate.

2.3 Retrieving Fingerprints from the Database

Match score values may vary significantly depending on the quality of the fingerprint impressions. The quality is impacted by the photometric and geometric variations in the images. This variation can cause the index code of a query image to be different from its corresponding model index code. However, the quantization function ensures that a majority of the individual elements constituting these two index codes are the same. During identification, when a potential list of candidate matches has to be determined, the Hamming distance can be used to find model index codes located in the neighborhood of the query index code. The size t of the neighborhood depends on the expected error rate between the query and model index codes, and is estimated empirically. One straightforward implementation of the search process is presented below (note that step 1 adds to the computational overhead of the search process; however, Hamming distances can be computed rapidly thereby improving overall response time, since the number of fingerprint-to-fingerprint comparisons is reduced).

Algorithm for searching the model index codes

Let $\boldsymbol{\mathcal{I}}_q$ be the index code of the query image and $\boldsymbol{\mathcal{I}}_{x_i}$ be the index code of image x_i from the database.

1. Compute the Hamming distances $d_i = D_h(\boldsymbol{\mathcal{I}}_q, \boldsymbol{\mathcal{I}}_{x_i})$ for $i = 1, \ldots, M$.
2. Retrieve all x_i such that $d_i \leq t$.

More efficient algorithms for ordering and searching the model codes can be designed to minimize the number of computations. Clearly, if the model index codes are well scattered in terms of Hamming distance, the number of retrieved fingerprints will be small. In the ideal case, these fingerprints will pertain to the same identity as that of the query index code.

2.4 Properties of the Index Space

To ensure unique index codes and the ability to handle errors in coding, the index space has to be large and sparse. The size of the index space can be increased indefinitely by using a larger n (the number of reference fingerprints). However, a n value that is greater than, for example, 5% of the database will defeat the purpose of indexing. Larger value of b also lead to a larger index space. For example, for $n = 20$ and $b = 2$ the number of possible index codes is $2^{20} \approx 10^6$, while for $b = 3$ this number is $3^{20} \approx 3.5 \times 10^9$. However, the value of b cannot be changed arbitrarily because it depends on the distribution of the impostor match scores (see section 2.2).

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

The sparsity of the index space can be measured in terms of n, b, and t. The Hamming bound [8] provides a theoretical limit on the maximum number of unique model index codes such that the Hamming distance between an arbitrary query code and its associated model code is at most t, and all other model codes are at a distance greater than t. The Hamming bound is defined as:

$$A_b(n,t) \leq \frac{b^n}{\sum_{i=0}^{t} \binom{n}{i}(b-1)^i} \ .$$

For $n = 20$, $b = 3$, and $t = 6$, $A_3(20,6) \leq 1137$. For a database of size $M = 2,000$, if we were to assume that the model index codes achieve the Hamming bound and that no more than two fingerprints have the same index code, then only 0.1% of the database will be retrieved by the search process of the proposed method (this is called *penetration rate*). In practice, such an ideal distribution cannot be achieved. First, the distribution of the index codes created using the proposed method is "random" in nature and, therefore, the probability of achieving the Hamming bound is very small.[3] Second, the variance of the errors is large, which will further decrease the Hamming bound. For example, $A_3(20,10) \leq 10$ resulting in a 10% penetration rate.[4]

3 Experiments

In this section we present the implementation details of the proposed indexing method using ternary index codes (i.e., $b = 3$). Two thresholds, T_1 and T_2 are used to split the domain of the match score into three regions. The values of the thresholds are relative to the mean value, μ, and the standard deviation, σ, of the available impostor scores: $T_1 = \mu - \sigma/2$ and $T_2 = \mu + \sigma/2$ (Figure 3). Therefore, the index code $\boldsymbol{\mathcal{I}}_x = \{i_1^x, i_2^x, ..., i_n^x\}$ is obtained as $i_j^x = \begin{cases} 0, & \text{for } s_j^x < T_1 \\ 1, & \text{for } T_1 \leq s_j^x \leq T_2 \\ 2, & \text{for } s_j^x > T_2 \end{cases}$.
The search process extracts all model index codes whose Hamming distance from the query index code is smaller than a certain value t.

Two fingerprint data sets were used to test the proposed methods. The first data set is the NIST special database 4 (NIST-4) [13], which is the database of choice for evaluating fingerprint classification algorithms. It contains inked fingerprints and exhibits large variations in quality. The database consists of the images of 2,000 fingers with two impressions per finger. The second dataset was taken from the West Virginia University (WVU) multimodal biometric database [4]. A subset of 2400 images corresponding to 800 fingers (three impressions per finger) were used. This dataset contains impressions of the index and thumb

[3] Furthermore, an algorithmic way to design codes that achieve the Hamming bound does not exist [8].

[4] This is assuming that the fingerprints from the database are uniformly distributed over the space of unique model index codes.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

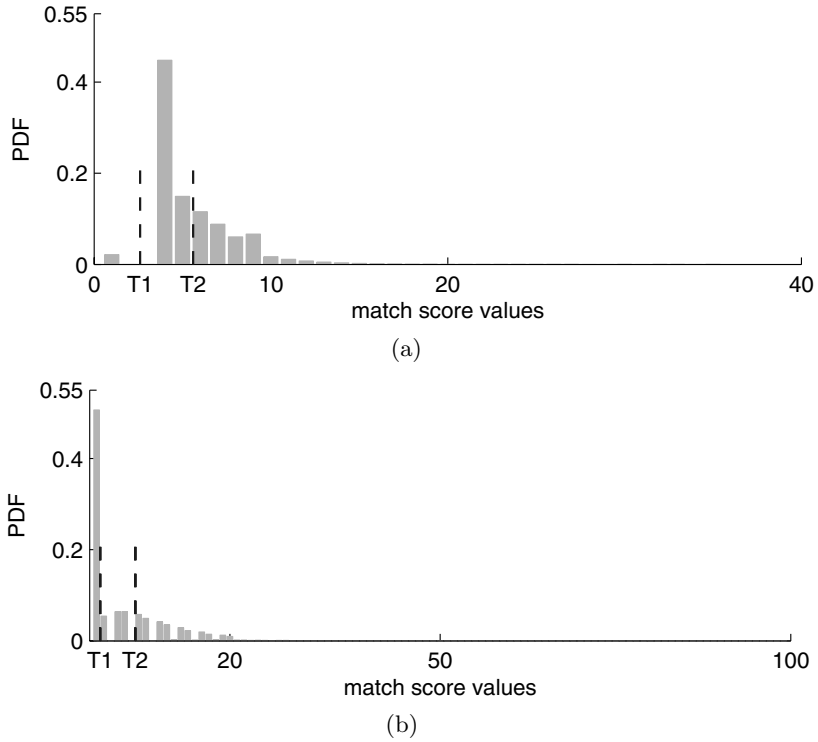

Fig. 3. Distribution of impostor scores and position of the thresholds. (a) NIST-4 data, $T_1 = 3, T_2 = 5$. (b) WVU data, $T_1 = 2, T_2 = 6$.

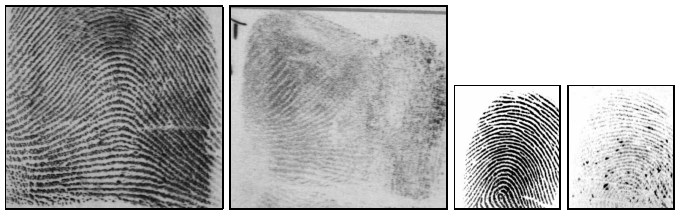

Fig. 4. Sample images from the two datasets. (a) NIST-4, good quality, (b) NIST-4, poor quality, (c) WVU, good quality, (d) WVU, poor quality.

fingers of both hands of a subject obtained using a SecuGen Hamster scanner. Figure 4 presents examples of fingerprint images from the two datasets. The match scores were generated using the VeriFinger SDK by Neurotechnology. Several values of n were used in our experiments ranging from $n = 50$ to $n = 250$ in order to explore the amount of diversity needed to select "good" reference fingerprints.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

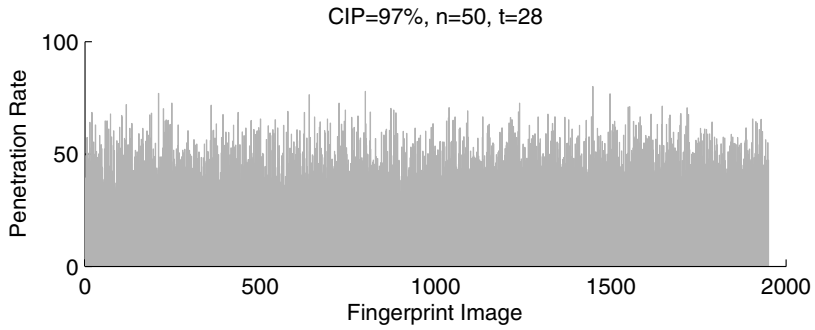

Fig. 5. Penetration rates of all images in the NIST-4 dataset for $n = 50$ and $t = 28$

4 Results

We evaluate the indexing performance in terms of Correct Index Power (CIP) [3] and penetration rate (R_p): $CIP = \frac{N_c}{M}$ and $R_p = \frac{N_a}{M}$, where N_c is the number of images that are correctly indexed, N_a is the number of images that are retrieved during a single identification operation, and M is the number of images in the data set. A query image is said to be correctly indexed, if one of the retrieved model index codes corresponds to the correct identity. In the experiments, one impression of each finger was used to populate the database (training set) while the other impressions of the finger were used to evaluate the performance of the proposed indexing method (test set). Cross-validation was performed by cycling through the multiple impressions of a finger whilst composing the training set. The penetration rate of all images in the test set of one particular experiment involving the NIST-4 database with $n = 50$ and $t = 28$ is shown in Figure 5. Figure 6 presents the CIP as a function of the penetration rate, which was averaged over all test images. For a 50% penetration rate the best CIP that was achieved was ∼98% for the NIST-4 dataset (n=50) and ∼99% for the WVU dataset (n=250). This is a promising result given that only match scores (rather than complex image features) have been used to facilitate indexing.

After a certain point, the indexing performance on the NIST-4 dataset decreases with increasing values of n. A possible explanation for this result is that the variability of the fingerprints in this dataset is so large that the use of a small set of reference fingerprints creates sparse index codes. In this case, increasing n beyond a certain value cannot add new variations and, therefore, will worsen the sparsity of the index space. Note that longer index codes usually leads to a larger number of errors. In contrast, the overall variability of the fingerprints in the WVU dataset may be small (due to the smaller size of this dataset and/or the nature of the fingerprints in it). Therefore, increasing the number of reference fingerprints (from 50 to 250) adds variability and a sparser index space is created.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

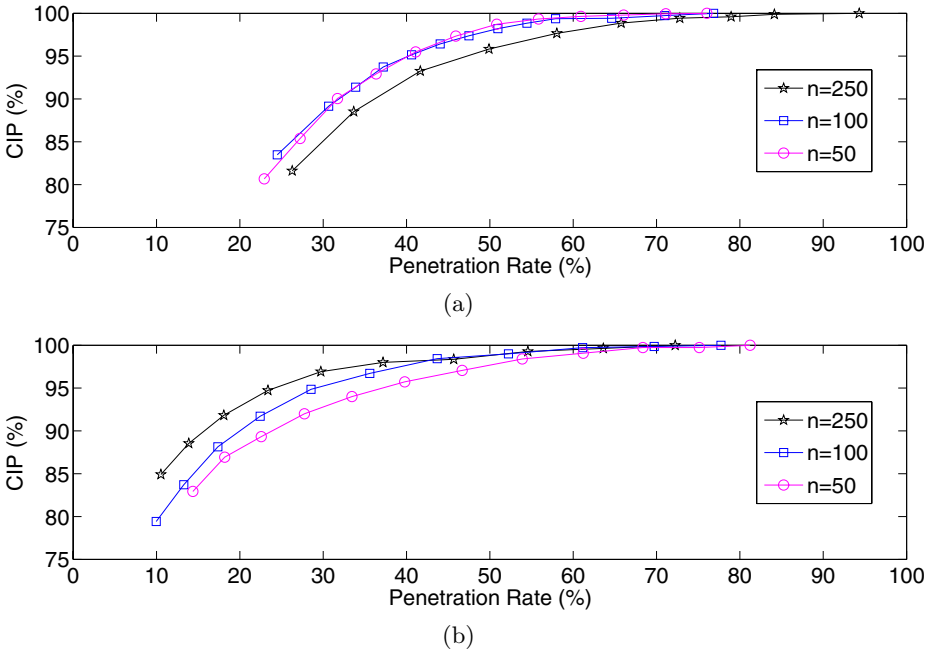

Fig. 6. Retrieval performance for different values of n. (a) NIST-4 dataset. (b) WVU dataset.

5 Summary and Future Work

In this paper, a new coding scheme for indexing fingerprint databases was presented. The proposed scheme relies on the use of a small set of reference fingerprints to generate a compact index code for labeling an arbitrary fingerprint. To handle noisy queries, candidate model index codes are identified using Hamming distances. The performance of the proposed scheme was evaluated on two fingerprint datasets. Experimental results demonstrate the efficacy of the scheme and its robustness to noisy query prints. However, there is plenty of room for improvement: (a) Estimating appropriate threshold values is critical for minimizing the number of errors in the index code caused by noisy images. We are exploring methods to automatically derive the discretization function based on the match score distributions associated with a matcher. (b) The selection of the reference fingerprints is important for obtaining a sparsely populated index space. Imposing further constraints on the selection mechanism for reference fingerprints may result in better sparsity. (c) The relationship between n and M for efficient indexing can be explored by conducting experiments on larger datasets. Indeed, it may be possible to derive upper bounds on indexing accuracy based on score distributions. We are currently evaluating the proposed scheme on other biometric traits.

Proceedings of S+SSPR Workshop, LNCS 5342, pp. 765–774, (Orlando, USA), December 2008.

References

1. International Biometric Group, Henry Classification System (2003), http://www.biometricgroup.com/HenryFingerprintClassification.pdf
2. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint Identification Using Delaunay Triangulation. In: International Conference on Information Intelligence and Systems (ICIIS), pp. 452–459 (1999)
3. Bhanu, B., Tan, X.: Fingerprint Indexing Based on Novel Features of Minutiae Triplets. IEEE Trans. Pattern Analysis and Machine Intelligence 25(5), 616–622 (2003)
4. Crihalmeanu, S., Ross, A., Schuckers, S., Hornak, L.: A Protocol for Multibiometric Data Acquisition, Storage and Dissemination. Technical Report, WVU, Lane Department of Computer Science and Electrical Engineering (2007)
5. Feng, J., Cai, A.: Fingerprint Indexing Using Ridge Invariants. In: International Conference Pattern Recognition, vol. 4, pp. 433–436 (2006)
6. Germain, R.S., Califano, A., Colville, S.: Fingerprint Matching Using Transformation Parameter Clustering. IEEE Computational Science & Engineering 4(4), 42–49 (1997)
7. Jain, A., Pankanti, S.: Automated Fingerprint Identification and Imaging Systems. In: Lee, H.C., Gaensslen, R.E. (eds.) Advances in Fingerprint Technology, 2nd edn. CRC Press, Boca Raton (2001)
8. Lin, S., Costello, D.J.: Error Control Coding, 2nd edn. Prentice-Hall, Inc., Upper Saddle River (2004)
9. Lumini, A., Maio, D., Maltoni, D.: Continuous Versus Exclusive Classification for Fingerprint Retrieval. Pattern Recognition Letters 18(10), 1027–1034 (1997)
10. Maeda, T., Matsushita, M., Sasakawa, K.: Identification Algorithm Using a Matching Score Matrix. IEICE Trans. Information and Systems, Special Issue on Biometric Person Authentication, The Institute of Electronics, Information and Communication Engineers 84(7), 819–824 (2001)
11. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, New York (2003)
12. Mhatre, A., Palla, S., Chikkerur, S., Govindaraju, V.: Efficient Search and Retrieval in Biometric Databases. In: SPIE Biometric Technology for Human Identification II, pp. 265–273 (March 2005)
13. Watson, C.J., Wilson, C.L.: NIST Special Database 4, Fingerprint database: Users' Guide. U.S. National Institute for Standards and Technology (1992), http://www.nist.gov/data/WebGuide/SD_4/FingerprintDB_4.htm