Journal of Information Assurance and Security 4 (2009) 117-123

Online fingerprint identification with a fast and distortion tolerant hashing

Hoi Le[1], The Duy Bui[2]

[1]Icis Laboratory, University of Calgary
2500 University Drive, Calgary, AB, Canada
leh@ulcagary.ca (*)

[2]Human Machine Interaction Laboratory, Faculty of Information Technology,
College of Technology, Vietnam National University, Hanoi
144 Xuan Thuy, Cau Giay, Hanoi
duybt@vnu.edu.vn

***Abstract*.** National ID card, electronic commerce, and access to computer networks are some scenarios where reliable identification is a must. Existing authentication systems relying on knowledge-based approaches like passwords or token-based such as magnetic cards and passports contain serious security risks due to the vulnerability to engineering-social attacks and the easiness of sharing or compromising passwords and PINs. Biometrics such as fingerprint, face, eye retina, and voice offer a more reliable means for authentication. However, due to large biometric database and complicated biometric measures, it is difficult to design both an accurate and fast biometric recognition. Particularly, fast fingerprint indexing is one of the most challenging problems faced in fingerprint authentication system. In this paper, we present a specific contribution by introducing a new robust indexing scheme that is able not only to fasten the fingerprint recognition process but also improve the accuracy of the system.

Keywords: fingerprint hashing, fingerprint authentication, error correcting code

1. Introduction

The requirement for reliable identification in the development of digital world has been a great motivation for research in biometrics. National ID card, electronic commerce, and access to computer networks are some scenarios where declaration of a person's identity is crucial. Currently existing authentication systems relying on knowledge-based approaches like passwords or token-based such as magnetic cards and passports are recently pointed out several disadvantages. With the increasing computational ability of computers, passwords and PINs are shown to be vulnerable by engineering-social attacks [16]. Moreover, a more serious security risk lies in the easiness of sharing or compromising passwords and PINs. Once passwords or PINs are compromised, traditional methods cannot distinguish between authorized user and impersonator. On the other hand, biometrics has a special characteristic that user is the key; hence, it is not easily compromised or shared. Therefore, biometrics offers means of reliable personal authentication that can address these problems and is gaining citizen and government acceptance.

To present, although fingerprint authentication systems have achieved significant progress, there are still a number of research issues that need to be addressed to improve the system efficiency, especially for online systems [7]. Almost all existing systems rely on full search based verification which is so tedious, time-consuming, and expensive that it does not meet the performance requirements of the new applications. By performance, we mean the accuracy and lookup speed of a fingerprint identification system. For a small database, the approach to exhaustively match a query fingerprint against all the fingerprints in database [25] is acceptable. For a large database, however, it is not desirable in practice without an effective fingerprint indexing scheme. Currently, there are two main approaches to reduce the response time of online identification: classification [5][18] and indexing techniques [2][3][6][11][13][20].

Another consideration for fingerprint indexing algorithm is fingerprint distortion. There are two kinds of distortion: transformation distortion and system distortion. In particular, due to fingerprint scanners can only capture partial fingerprints; some fingerprint features may be missing during acquisition process. These distortions of fingerprint including fingerprint features missing cause several problems: (i) the number of fingerprint features available in such prints is few, thus reducing its discrimination power; (ii) loss of singular points (core and delta) is likely [17]. Therefore, a robust indexing algorithm independent of such global features is required.

In this paper, we propose an approach for online fingerprint identification with a fast and distortion tolerance hashing scheme. The hashing scheme, which can perform on any localized features such as minutiae or triplet, can achieve high efficiency in hashing performance. To simplify our system, we avoid minutiae alignment process by using the triplet feature introduced by Tsai-Yang Jea et. al. [17]. These triplet features are showed to be invariant to fingerprint transformations. By invariance we mean two triplet features are considered to be the same if the difference between their measurements is in some certain threshold. Therefore, to produce the same indexing result for two triplet features that are invariant, we present a definition of codeword for those fingerprint triplet feature points and perform indexing on

(*) *This work is accomplished when author was still working at department of Computer Science, Vietnam National University, Hanoi*

Received June 10, 2009

those codewords. This is one of our main contributions. The second contribution lies in the use of a randomness extractor as indexing function in order to reduce the number of candidates for matching stage.

The paper is organized as follows. Section 2 introduces different kinds of features which are often used in fingerprint matching. We then review related work on fingerprint matching in Section 3. In Section 4, we propose an error correction code for fingerprint feature points which are used in our hashing scheme, which is described in Section 5. We present experiment results of our scheme in section 6.

2. Fingerprint features

The fingerprint features which are extracted from fingerprint images can be categorized as global features, minutiae features and inter-minutiae features [9].

Core and delta points are global features which are distinguished by their position (see Figure 1). Core points are found where ridges of the fingerprints form a loop. On the other hands, delta points are the points where ridges surround an imaginary triangle.

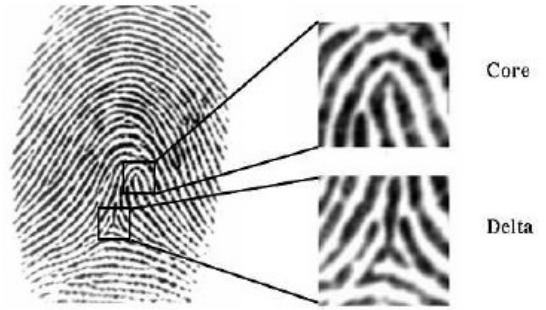

Figure 1. Core and Delta points.

Minutiae are probably the most important fingerprint features. Minutiae are the various ridge discontinuities of a fingerprint. There are two types of widely used minutiae which are bifurcations and endings (Figure 2). Minutiae are characterized by their position and direction (tangent direction at the appropriate ridge). Minutia based algorithms might not be affected by translation, but still face many difficulties with rotation.

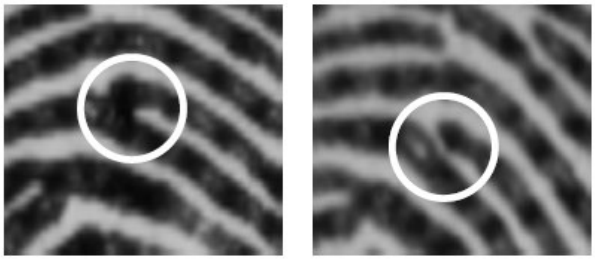

Figure 2. Ridge bifurcation (left) and ridge endings (right).

There are several types of inter-minutiae features. The first one is the Euclidean distance between two minutiae. The second one is the number of ridges between two minutiae points. A more sophisticated inter-minutiae feature is proposed in [17]. The feature, which is called secondary feature, is a vector of five-elements, as can be seen in Figure 3. A secondary feature is constructed from each minutia $M(x, y, \theta)$ and its two nearest neighbors $N_0(x_0, y_0, \theta_0)$ and $N_1(x_1, y_1, \theta_1)$, where (x, y), (x_0, y_0) and (x_1, y_1) are the position of *M*, *N0* and *N1* respectively, and θ, θ_0, and θ_1 are the orientation of *M*, *N0* and *N1* respectively. The secondary feature is in the form of $S(r_0, r_1, \varphi_0, \varphi_1, \delta)$ in which r_0 and r_1 are the Euclidean distances between the minutia M and its neighbors N_0 and N_1 respectively, φ_0 and φ_0 are the orientation difference between M and N_0, N_1 respectively, and δ represents the acute angle between the line segments MN_0 and MN_1. This kind of secondary feature is suitable for matching partial fingerprints because it does not depend on global features, e.g. core and delta. It is also invariant to orientations, which is one of main distortions to fingerprints.

Different from other approaches, Jain et al. [15] have presented a filter-based representation of fingerprints, which can then be used to obtain high accuracy in fingerprint matching.

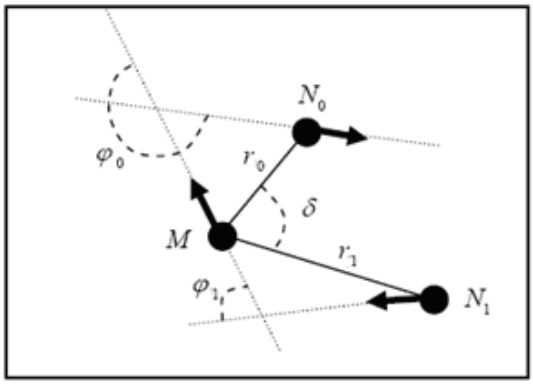

Figure 3. A secondary feature formed by a minutia M and its neighbors N_0 and N_1.

3. Related Work

A common approach to fingerprinting is to exhaustively match a query fingerprint against all the fingerprints in a small database based on extracted features [17] [25]. For a given pair of minutiae pair $p(x_p, y_p, \theta_p)$ and $q(x_q, y_q, \theta_q)$ it is said that p matches q if q is within the tolerance area of p. Let us consider $p(r_p, \Phi_p, \theta_p)$ and $q(r_q, \Phi_q, \theta_q)$ in polar coordinates. p matches q if for given threshold functions $Thld_r(.)$, $Thld_\phi(.)$, and $Thld_\theta(.)$, $|r_p - r_q| \le Thld_r(r_p)$, $|\phi_p - \phi_q| \le Thld_\phi(\phi_p)$ and

$\left|\theta_p - \theta_q\right| \le Thld_\theta\left(\theta_p\right)$. In order to tolerate against distortions like translation, rotation and missing features, Jea and Govindaraju have proposed a kind of secondary feature [17], which is described in previous section. Now, two secondary features $S_i(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$ and $S_j(r_{j0}, r_{j1}, \varphi_{j0}, \varphi_{j1}, \delta_j)$ match each other, if $\left| r_{i0} - r_{j0} \right| \le Thld_r(ri0)$, $\left| r_{i1} - r_{j1} \right| \le Thldr(r_{i1})$, $\left| \varphi_{i0} - \varphi_{j0} \right| \le Thld_\theta(\varphi_{i0})$, $\left| \varphi_{i1} - \varphi_{j1} \right| \le Thld_\theta(\varphi_{i1})$, and $\left| \delta_i - \delta_j \right| \le Thld_\delta(\delta_i)$. To obtain the optimal pairing of features of two fingerprints, Minimum Cost Flow is used [17].

It is not desirable in practice to exhaustively match a query fingerprint against all the fingerprints in a large database, especially for online identification. Therefore, an effective fingerprint indexing scheme is needed here. Currently, there are two main approaches to reduce the response time of online identification: classification and indexing techniques. Traditional classification techniques (e.g. [5], [18]) attempt to classify fingerprint into five classes: Right Loop, Left Loop, Whorl, Arch, and Tented Arch. Due to the uneven natural distribution, the number of classes is small and real fingerprints are unequally distributed among them: over 90% of fingerprints fall in to only three classes (Loops and Whorl) [20]. This is resulted in the inability of reducing the search space enough of such systems. Indexing technique performs better than classification in terms of the size of space need to be searched. Fingerprint indexing algorithms select most probable candidates and sort them by the similarity to the query. Many indexing algorithms have been proposed recently. A.K. Jain et al [13] use the features around the core point of a Gabor filtered image to realize indexing. Although this approach makes use of global information (core point) but the discrimination power of just one core is limited. In [2], the singular point (SP) is used to estimate the search priority which is resulted in the mean search space below 4% the whole dataset. However, detecting singular point is a hard problem. Some fingerprints even do not have SPs and the uncertainty of SP location is large [20]. Besides, several attempts to account for fingerprint indexing have shown the improvement. R. Cappelli et al. [6] proposed an approach which reaches the reasonable performance and identification time. R.S Germain et al. [11] use the triplets of minutiae in their indexing procedure. J.D Boer et al. [3] make effort in combining multiple features (orientation field, FingerCode, and minutiae triplets). In [21], Liu et al. has proposed an indexing approach based on singular points to fasten the indexing process. Their approach, however, cannot deal with fingerprints having no singular points.

4. An error correction code for fingerprint feature points

4.1. Error correction code

Error correction code (ECC) [23] is an efficient technique which is first developed for noise tolerance purpose in data communication and data storage. In general, an ECC can be defined as follow.

For a metric space M and a metric $\mathbf{d}$, a code is defined as a subset of K elements $C = \{w_1, ..., w_K\} \subset M$. The set C is also called codebook. Each of its elements w_i is called a codeword. According to the metric $\mathbf{d}$, the (minimum) distance of a code is defined as the smallest distance d between two distinct codewords.

Over a codebook C, we can define a pair of encoding-decoding functions ($\mathbf{C}$,$\mathbf{D}$). The encoding function $\mathbf{C}$ is an injective map for the elements of some domain of size K to the codewords in C. The decoding function $\mathbf{D}$ outputs pre-image $\mathbf{C}^{-1}[w_k]$ of an element $w \in M$, in which w_k is a codeword that minimizes the distance $\mathbf{d}[w, w_k]$.

The *error correcting distance* is the largest radius t such that for every element w in M there is at most one codeword in the ball of radius t centered on w. For integer distance functions we have $t = \left\lfloor {(d-1)}/{2} \right\rfloor$. A standard shorthand notation in coding theory is that of a (M, K, t)-code.

There are some well-known ECC schemes such as Hamming Code, Reed-Solomon Code, etc. An example of their applications is in data communication where the channel is noisy which probably causes the original message corrupted when it is received at the receiver. In order to transmit the original message successfully, it is added some redundant information before transmitting. This process is called Encoding using the encoding function of one appropriate ECC. And, the original message appended with this redundant information is the codeword for such message. This codeword is transmitted and received at the other end of the noisy channel. It can be reconstructed to the original message successfully if there are at most t errors occurred during transmission process using decoding function.

Nowadays, ECC has been widely applied in various areas, for example, in cryptography it is used to guarantee data integrity. In this paper, we will introduce a new ECC scheme in the next sub-section for the same error tolerance purpose but in a particular noisy data like fingerprint.

4.2. Our modified error correction code

In this sub-section, we define a new ECC for fingerprint data. In our scheme, fingerprint template is represented in feature points. In particular, we treat a feature point x as a corrupted codeword caused by measuring process, and try to correct it to the correct codeword by using an encoding function $C(x)$. Below is the formal definition of our encoding function. Note that, we do not define the decoding function for our ECC because it is not being used in our indexing scheme. Moreover, to tailor the ECC for our indexing scheme, output of the encoding function is a set of three codewords instead of only one codeword as in the general ECC definition.

Definition 1: Let $x \in R, t \in R$, we define $C(x)$, the encoding function, as $(q(x-t), q(x), q(x+t))$ where $q(x)$ is a quantization function of x with quantization step t. Thus, the output of $C(x)$ is (c_{x-t}, c_x, c_{x+t}) which is called the *codeword set* of x, where $c_x = q(x)$ is called the codeword of x.

Lemma 1: Let $x \in R$, given a real number $t \in R$. For every y such that $|x-y| \le t$, then the codeword of y $c_y = q(y)$ equals to one of three elements in the *codeword set* of x.

Lemma 1 can be proved easily by some algebraic transformations.

In our approach, we generate the codeword set for every dimension of a template feature point q and only the codeword for every dimension of a query point p.

5. Codeword-based hashing scheme

We now present a fingerprint hashing scheme based on codewords of feature points. The key idea is summarized as follows: first to "correct" each feature point after extracting to its codeword set to achieve error tolerant purpose, then to use each element of that codeword set as the input of a special hash function which is able to produce uniformly distributed hash values.

5.1. Our approach

Informal description. Fingerprint features stored in database can be considered as a very large set. Moreover, the distribution of fingerprint feature points is not uniform and unknown [7]. One of the traditional techniques to reduce the searching time in a database is hashing. However, if the input set is not uniform, this traditional hashing becomes inefficient because it results in a hash table of which the size of hash entry is unlikely uniform, therefore the average expected searching time is not much lower than the case where linear search strategy is used. Therefore, in order to improve the search speed, we want to design an indexing scheme such that for the large and not uniformly distributed input sets of feature points, it can produce uniformly distributed hash values for searching process. To match this requirement, there is one technique called random extractor which is widely used in cryptography and other purposes related to randomness generation [24]. A random extractor is a function that takes an input a high entropy and not uniform source and generates an output shorter and uniformly distributed. A randomized hash function is a construction of random extractors. Following is our description of our scheme which builds such randomized hash function to perform indexing on fingerprints.

Our randomized hash function includes a randomized permutation function and a standard hashing function. First, bits of the input (in binary form) are permuted by a randomized permutation and then the hash of it is computed by a compression function e.g. SHA. The basic idea so far is for any given set of inputs, this algorithm will scatter the inputs among the range of the output of the function well based on a random permutation so that the output will be distributed more randomly. For detailed description please refer to the formal description part below.

To ensure the error tolerant property, we perform hashing on the codeword instead of the feature point itself. Follow the lemma 1 we have the query point y and the template point x are matched if and only if codeword of y belongs in the codeword set of x.

Formal description. We set up our indexing scheme as follows.

1. *Choose L dimensions from a dimension set $(1, 2, ..., D)$ of a feature point.*
2. *Choose a tolerant threshold t and an appropriate metric function l_d for each selected dimension.*

For each template point p:

1. *Generate the codeword set for p_d. These values are mapped to binary strings.*
2. *Binary strings in the codeword set of selected dimensions are then concatenated to form L^3 binary strings m_i for $i = 1, ..., L^3$ where L is the number of selected dimensions.*
3. *Each m_i is padded with zero bits to form a n – bit message m_i.*
4. *Generate a permutation π of $\{0,1\}^n$ for m_i by using a random permutation function.*
5. *m_i is permuted by π and the resulting message is hashed using a compression function such as SHA.*

$$Let\ sh_i = SHA(\pi(m_i))\ for\ i = 1, ..., L^3$$

6. *There are at L^3 hash values for p. These values can be stored in the same hash table as well as separate ones for respectively dimensions.*

For query point q:

1. *Compute the codeword for every selected dimension of q. Then map this codeword to a binary string.*
2. *Binary strings of codewords of selected dimensions are then concatenated.*
3. *Perform steps 3 to 5 as for a template point. Note that there is only one binary string for a query point q*
4. *Return all the points which are sharing the identical hash value with q.*

For query evaluation, all candidate matches are returned by our hashing for every query feature point. Hence, each query fingerprint is treated as a bag of points, simulating a multi-point query evaluation. To do this efficiently, we use an identical framework as Ke et al. [19], in that, we maintain two auxiliary index structure –File Table (FT) and Keypoint Table (KT) – to map feature points to their corresponding fingerprint; an entry in KT consists of the file ID (index location of FT) and feature point information.

The template fingerprints that have points sharing identical hash values (collisions) with the query version are then ranked by the number of similarity points. Only top T templates are selected for identifying the query fingerprint by matching. Thus, the search space is greatly reduced. The candidate selection process requires only linear computational cost so that it can be applied for online interactive querying on large image collections.

5.2. Analysis

Space complexity

For an automate fingerprint identification, we must allocate extra storage for the hash values of template fingerprints.

Assume that the number of D-dimension feature points extracted from template version is N. With L selected dimensions, the total extra space required for one template is $O\left(L^3.N\right)$.

Thus, the hashing scheme requires a polynomial-sized data structure that allows sub-linear time retrievals of near neighbors as shown in following section.

Time complexity

On query fingerprint Y with N feature points in a database of M fingerprints, we must

- Compute the codeword for each feature point which takes $O\left(N\right)$ due to quantization hashing functions required constant time complexity.
- Compute the hash value which takes $O\left(time\left(f,x\right)\right)$ where f is the hash function used and $time\left(f,x\right)$ is the time required by function f with input x.
- Compute the similarity scores of templates that has any point sharing identical hash value with the query fingerprint, requiring time $O\left(\frac{M.N}{2^n}\right)$.

Thus the key quantity is $O\left(\frac{M.N}{2^n}+time\left(f,x\right)\right)$ which is approximately equivalent to $1/2^n$ computations of exhaustive search.

6. Experiments

We evaluate our method by testing it on Database DB1_A FVC2004 obtained by an optical sensor which is described in [22]. DB1_A database contains the partial fingerprint templates of various sizes. These images are captured with a resolution of 500dpi, resulting in 8 bit gray scale images.

We use the feature extraction algorithm described in [17] in our system. However, the authors do not mention in detail how to determine the threshold of error tolerances. Therefore, in our experiments, we assume that the error tolerance threshold is fixed t for cases of features. Note that this assumption may make the implementation in optimal as expected. Distance function used is Euclidean metric.

Table I shows the result in Correct Index Power (CIP), which is defined as the percentage of correctly indexed queries based on the percentage of hypotheses that need to be searched in the verification step. Although our implementation is not optimal, the scheme still achieves good CIP result. As can be easily seen, the larger search percentage is, the better results are obtained. It indicates that the optimal implementation can improve the result performance.

Correct Index Power				
Search Percentage	5%	10%	15%	20%
CIP	80%	87%	94%	96%

Table 1. Correct Indexing Power of our algorithm

Compared with some published experiments in the literature, at the search percentage 10%, [15] comes up with 84.5% CIP and [21] reaches a result of 92.8% CIP, while our approach reaches 87% (see Figure 4). At the search percentage 20%, [21] reaches a result of about 94% CIP, while our approach reaches 96%. Moreover, the approach in [21] cannot deal with the fingerprints that have no singular points.

Unlike previous works, our scheme is reasonably simpler and by adjusting *t* carefully, it is promising that our scheme will reach 100% CIP with low search percentage.

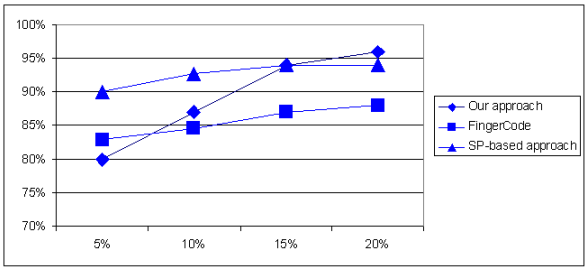

Figure 4. CIP of our approach compared to FingerCode[15] and SP-based approach [21].

7. Conclusion

In this paper, we have presented a new robust indexing scheme on fingerprints which attempts to provide both accurate and fast indexing. However, there is still some works need to be considered further in order to make the system more persuasive and obtain the best optimal result such as studying optimal choices for parameter *t*. It is possible that the scheme would give a better performance with other representation of fingerprint templates. Besides, assuring privacy property for original fingerprint templates in an indexing scheme is another important problem that should also be considered.

References

[1] Bazen A. M. and Sabih H. Gerez. "Fingerprint matching by thin-plate spline modeling of elastic deformations". *Pattern Recognition*, issue 36, pages 1859–1867, 2003.

[2] Bazen A. M., Verwaaijen G. T. B, Garez S. H., Veelunturf L. P. J., and B. J. van der Zwaag. "A correlation-based fingerprint verification system". *ProRISC2000 Workshops on Circuits, Systems and Signal Processing*, pp. 205-213 2000.

[3] Boer J., Bazen A., and Cerez S. "Indexing fingerprint database based on multiple features". *ProRISC 2001 Workshop on Circuits, Systems and Singal Processing*, pp. 300-306, 2001.

[4] Brown L. *A survey of image registration techniques*. ACM Computing Surveys, 1992.

[5] Cappelli R., Lumini A., Maio D., and Maltoni D. "Fingerprint Classification by Directional Image Partitioning". *IEEE Trans. on PAMI*, volume 21, issue 5, pages 402-421, 1999.

[6] Cappelli R., Maio D., and Maltoni D. "Indexing fingerprint databases for efficicent 1 : n matching". *Sixth Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, 2000.

[7] R. Cappelli and D. Maltoni, "On the Spatial Distribution of Fingerprint Singularities", *IEEE Transactions on Pattern Analysis Machine Intelligence*, volume 31, no.4, pp.742-748, April 2009.

[8] Choudhary A. M. and Awwal A. A. S. "Optical pattern recognition of fingerprints using distortion-invariant phase-only filter". *SPIE, Photonic devices and algorithms for computer*, volume 3805, pages 162–170. (20), 1999.

[9] Gerald Eckert, Soenke Müller, Torsten Wiebesiek, "Efficient Minutiae-Based Fingerprint Matching", *IAPR Conference on Machine Vision Applications MVA*, Japan, pp. 554--557, 2005.

[10] Fingerprint verification competition. http://bias.csr.unibo.it/fvc2002/.

[11] Germain R., Califano A., and Colville S.. "Fingerprint matching using transformation parameter clustering". *IEEE Computational Science and Eng.*, volume 4, issue 4, pages: 42-49, 1997.

[12] Gonzalez, Woods, and Eddins. *Digital Image Processing*. Prentice Hall, 2004.

[13] Anil Jain, Arun Ross, and Salil Prabhakar. "Fingerprint matching using minutiae texture features". *International Conference on Image Processing*, pages 282–285, 2001.

[14] Anil Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. "Filterbank-based fingerprint matching". *Transactions on Image Processing*, volume 9, pages 846-859, 2000.

[15] Jain A.K., Prabhakar S., Hong L., and Pankanti S. "FingerCode: a filterbank for fingerprint representation and matching". *Computer Vision and Pattern Recognition IEEE Computer Society Conference*, volume 2, pages 187-193, 1999.

[16] Jea T., Chavan V. K., Govindaraju V., and Schneider J. K. "Security and matching of partial fingerprint recognition systems". *SPIE*, pages 39–50, 2004.

[17] Jea Tsai-Yang, and Govindaraju Venu. "A minutia-based partial fingerprint recognition system". *Pattern Recognition*, volume 38, issue 10, pages 1672-1684, 2005.

[18] Karu K. and Jain A.K.. "Fingerprint Classification". *Pattern Recognition*, volume 18, issue 3, pages 389-404, 1996.

[19] Ke Y., Sukthankar R. and Huston L. "An efficient parts-based near duplicate and sub-image retrieval system". *MM International Conference on Multimedia*, pages 869–876, 2004.

[20] Liang X., Asano T., and Bishnu A. "Distorted Fingerprint indexing using minutiae detail and delaunay triangle". *3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'06)*, pages 217-223, 2006.

[21] Tong Liu, Guacai Zhu Chao Zhang and Pengwei Hao. "Fingerprint Indexing Based on Singular Point Correlation". *ICIP05*, volume 3, pages 293-296, 2005.

[22] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain. FVC2004: "Third Fingerprint Verification Competition". *Proc. International Conference on Biometric Authentication (ICBA)*, pp. 1-7, Hong Kong, July 2004.

[23] Purser Micheal, *Introduction to Error Correcting Codes*. Artech House Publishers, 1994.

[24] Nandakumar K. and Jain A. K. "Local correlation-based fingerprint matching". *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 503–508, 2004.

[25] Nist fingerprint vendor technology evaluation (http://fpvte.nist.gov/).

[26] Bolle Ruud, Connell J. H., Pankanti S., Ratha N. K., and Senior A. W. *Guide to Biometrics*. Springer Verlag, 2003.

[27] Salil Vadhan. *Randomness Extractors and their Cryptography Applications*. Tutorial in: Theory of Cryptography Conference, 2008.

Author Biographies

Hoi Le. Hoi Le was born in July 11th, 1983 in Vietnam. She is now a PhD student in Computer Science, University of Calgary, Canada. Before this, she earned her Bsc degree at College of Technology, Vietnam National

University, and after that worked there as an associate researcher and lecturer.

The Duy Bui. The Duy Bui got his Bachelor degree in Computer Science at University of Wollongong in 2000, and his PhD in Computer Science at University of Twente in 2004. From 2004 to present, he works at College of Technology, Vietnam National University, Hanoi as a lecturer. His main research interests are Human Computer Interaction, Virtual Reality and Multimedia.