Parallel Processing of the Fuzzy Fingerprint Vault based on Geometric Hashing

Seung-Hoon Chae[1], Sung Jin Lim[1], Sang-Hyun Bae[2], Yongwha Chung[3] and Sung Bum Pan[1,4]
[1] Dept. of Information and Communication Engineering, Chosun Univ.
375, Seosuk-dong Dong-gu, Gwangju, 501-759 - Korea
[e-mail: ssuguly@gamil.com]
[2] Dept. of Computer Science and Statistics, Chosun Univ.
375, Seosuk-dong Dong-gu, Gwangju, 501-759 - Korea
[3] Dept. of Computer and Information Science, Korea Univ.
Chochiwon, Chungnam, 339-700 - Korea
[4] Dept. of Control, Instrumentation, and Robot Engineering, Chosun Univ.
375, Seosuk-dong Dong-gu, Gwangju, 501-759 - Korea
[e-mail: sbpan@chosun.ac.kr]
*Corresponding author: Sung Bum Pan

Received May 24, 2010; revised July 30, 2010; revised September 8, 2010; revised October 6; accepted October 7, 2010; published December 23, 2010

Abstract

User authentication using fingerprint information provides convenience as well as strong security. However, serious problems may occur if fingerprint information stored for user authentication is used illegally by a different person since it cannot be changed freely as a password due to a limited number of fingers. Recently, research in fuzzy fingerprint vault system has been carried out actively to safely protect fingerprint information in a fingerprint authentication system. In addition, research to solve the fingerprint alignment problem by applying a geometric hashing technique has also been carried out. In this paper, we propose the hardware architecture for a geometric hashing based fuzzy fingerprint vault system that consists of the software module and hardware module. The hardware module performs the matching for the transformed minutiae in the enrollment hash table and verification hash table. On the other hand, the software module is responsible for hardware feature extraction. We also propose the hardware architecture which parallel processing technique is applied for high speed processing. Based on the experimental results, we confirmed that execution time for the proposed hardware architecture was 0.24 second when number of real minutiae was 36 and number of chaff minutiae was 200, whereas that of the software solution was 1.13 second. For the same condition, execution time of the hardware architecture which parallel processing technique was applied was 0.01 second. Note that the proposed hardware architecture can achieve a speed-up of close to 100 times compared to a software based solution.

Keywords: Biometrics, fingerprint authentication, fuzzy fingerprint vault, geometric hashing, parallel processing

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2009-0086148). This research was financially supported by the Ministry of Education, Science Technology(MEST) and Korea Institute for the Advancement of Technology(KIAT) through the Human Resource Training Project for Regional Innovation.

DOI: 10.3837/tiis.2010.12.018

1. Introduction

In current society, it is possible to access information easily anytime and anywhere due to advances in communication media like the Internet. As accessing information has become simple, importance of user authentication and strong authentication system to protect personal information is increasing. However, for the password method that is supported in most of current applications, password length is not sufficient and users tend to use characters that are easy to remember within the password size that is supported for many applications rather than using random passwords. Therefore, passwords can be guessed or leaked easily [1]. Such problems in the user authentication based password method can be solved by using the biometric information of the user. However, if biometric information that has been stored for user authentication is used illegally by another person, serious problems may cause since changing it is impossible or limited, unlike a password.

In this paper, fingerprint which is evaluated as the most realistic alternative solution in terms of availability, accuracy and economy was selected among various biometrics [2]. Even in an user authentication system using the fingerprint, there are several problems. First, there is the disadvantage that fingerprint information input from the same finger of the same person does not match completely every time even though it is similar, unlike a password. In addition, it is difficult to change in case it is leaked because fingers of people are limited in number. Since a user uses identical biometric information for a variety of applications, leaked biometric information can be reused in all applications. Therefore, study to safely protect biometric information from illegal acquisition, falsification, and alteration is needed [3][4][5][6].

Recently, study on the fuzzy fingerprint vault which fuzzy vault theory is applied to fingerprint authentication has been carried out after Juels and Sudan proposed the fuzzy vault theory. Fuzzy fingerprint vault is a cryptology method which secret key and fingerprint information of the user are combined to obtain a secret key for only the right user. Real minutiae of the user is protected by creating a polynomial using the user's secret key and organizing the fingerprint template of the user with the user's real minutiae after creating the chaff minutiae randomly [7].

While study to simultaneously protect user fingerprint information and secret key of that user using fuzzy fingerprint vault is being reported [8][9], they cannot be realized because the alignment process was omitted due to absence of the fingerprint. To solve this problem, a method which geometric hashing technique is applied to the fuzzy fingerprint vault system was proposed [10][11][12][13]. The geometric hashing technique is an object authentication algorithm which object information is extracted and stored in a database and searched after being geometrically transformed [14]. And, recently a study which indicated that fingerprint fuzzy vault system is vulnerable to correlation attack was published [15]. Therefore a series of studies on the method of creating template of fingerprint fuzzy vault which is strong against correlation attack are in progress [16][17].

In this paper, we propose the hardware architecture for a fuzzy fingerprint vault system based on geometric hashing. The proposed architecture is performed by combining the software and hardware modules. The software module consists of modules for fingerprint minutiae information extraction, fingerprint template generation, fingerprint hash table generation and database storage. The hardware module consists of the matching module, verification module and memory to store the enrollment hash table and verification hash table.

The matching module compares the transformed minutiae of the enrollment hash table and transformed minutiae of the verification hash table. The verification module takes the role of the calculation according to the result of the matching module. In addition, we propose a hardware architecture which parallel processing technique is applied for high speed processing of the fuzzy fingerprint vault system. Software module is identical to the previously proposed hardware architecture. The hardware module consists of the memory for storing the enrollment hash table and verification hash table, matching modules and the verification module.

This paper is organized as follows. In Section 2, fuzzy vault and geometric hashing technique are described. Explanation for fingerprint fuzzy vault based on geometric hashing and the proposed hardware architecture are given in Section 3. Experiment result is given in Section 4 and conclusion for this paper is made in Section 5.

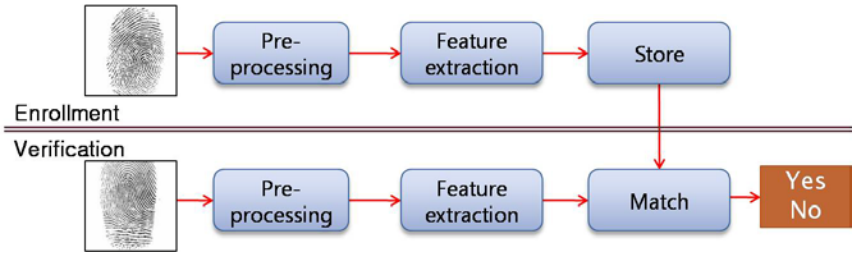


Fig. 1. Fingerprint authentication algorithm flowchart

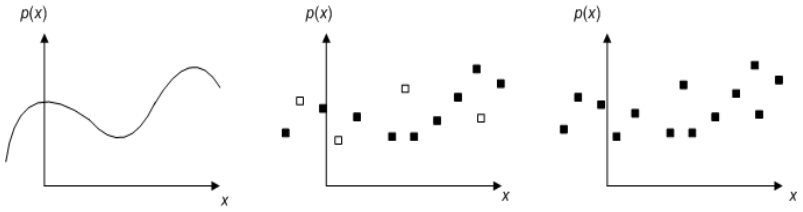
2. Background

A fingerprint authentication system is divided into the enrollment and verification processing as shown in **Fig. 1**. Offline enrollment processing is carried out in following order: Pre-processing, minutiae extraction and storage. And online verification processing verifies whether it is the correct person by comparing the minutiae extracted from the input image and the stored minutiae. However, serious problems will cause if such fingerprint information that is stored in the fingerprint authentication system is leaked or attacked. If fingerprint information is leaked from a fingerprint authentication system, there is a limit in changing a leaked fingerprint since each person may only use 10 fingerprints and a leaked fingerprint information cannot be re-registered.

2.1 Fuzzy Vault

Juels and Sudan's fuzzy vault scheme [7] is an improvement upon the previous work by Juels and Wattenberg [18]. Alice can place a secret value k (e.g., private encryption key) in a vault and lock(secure) it using an unordered set *A*. Bob, using an unordered set *B*, can unlock the vault(access κ) only if *B* overlaps with *A* to a great extent. The procedure for constructing the fuzzy vault is as follows: First, Alice selects a polynomial p of variable x that encodes k (e.g., by fixing the coefficients of *p* according to κ). She computes the polynomial projections, *p*(*A*), for the elements of *A*. She adds some randomly generated chaff points that do not lie on *p*, to arrive at the final point set *R*. When Bob tries to learn κ (i.e., finding *p*), he uses his own unordered set *B*. If *B* overlaps with *A* substantially, he will be able to locate many points in *R* that lie on *p*. Using error-correction coding, it is assumed that he can reconstruct *p* (and hence κ). The security of the scheme is based on the infeasibility of the polynomial reconstruction problem as shown in **Fig. 2** (i.e., if Bob does not know many points that lie on *p*, he cannot

feasibly find the parameters of p, hence he cannot access κ). Note that since this fuzzy vault can work with unordered sets(common in biometric templates, including fingerprint minutiae data), it is a promising candidate for biometric cryptosystem.



(a) Polynomial **(b)** Evaluation of the polynomial **(c)** Final vault list

Fig. 2. Polynomial and vault for the fuzzy vault

2.2 Geometric Hashing

In computer vision, objects are expressed as a set of features. Object features include points, lines, corners and edges. Object recognition refers to recognition of objects for which there is interest in the scene of an input image. In other words, objects for which there is interest in the scene of an input image can be recognized by storing the feature of the model which is the object in the database and searching for models in the database. An object recognition system like this is called a model based recognition system. Most model based recognition systems assume that object feature for the input image and feature of a single model stored in the database are identical and verifies the assumption. Such a process is carried out entirely for all models stored in the database. Therefore, processing time varies with the number of models stored in the database and processing time will become longer as the number of models are increased. To solve this problem, Lamdan and Wolfson introduced a geometric hashing algorithm based on indexing to object recognition [19].

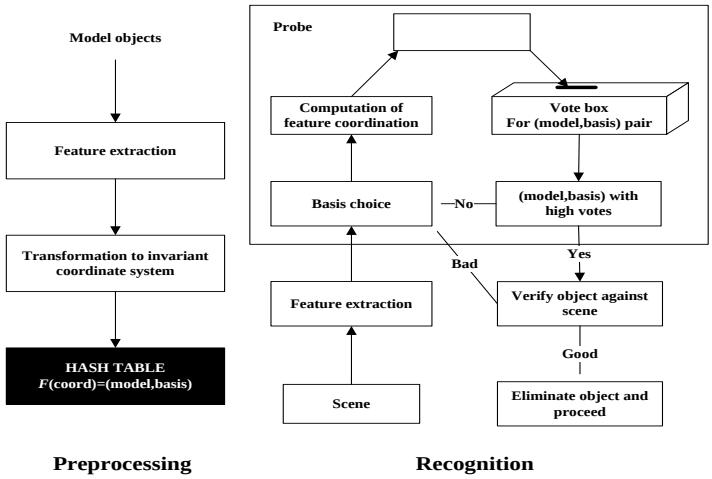


Fig. 3. Block diagram for geometric hashing

Geometric hashing algorithm generates an indexing function by considering the fact that geometric shape, which expresses an object does not change. It is an algorithm that searches a database efficiently by using this [14]. If geometric hashing algorithm is used in a model based

recognition system, effect on the processing time will be insignificant even if the number of models increase if the performance of the indexing function is acceptable. Geometric hashing algorithm consists of preprocessing and recognition processes as shown in **Fig. 3**. Preprocessing is carried out only once offline. First, the characteristic point which is expressed as an object point is stored in hash table form which includes a large amount of information due to geometric transformation to store in the database. For example, if *n* characteristic points of an object are stored in the database, one characteristic point is selected as a reference point and moved to the center of the coordinate axis by using geometric transformation such as rotation or translation along with a different characteristic point that forms a pair. The remaining characteristic points are converted using the same method and the converted coordinates are expressed with a method that is defined as bin for the hash table. reference point are selected like this for all *n* characteristic points and the remaining characteristic points are converted using the same method.

The recognition process, carried out online, extracts the characteristic point *S* of the object to recognize, selects one arbitrary characteristic point as a reference point and moves it to the center of the coordinate axis by using geometric transformation such as rotation or translation along with a different characteristic point that forms a pair. And the remaining characteristic points are converted using the same method. A table created like this increases the number of bins by comparing with the hash table that is created during preprocessing. Then, the reference points are selected again for all *n* characteristics and the bin with the highest score is recognized as the most similar object after converting the remaining characteristic points with the same method. For reference point selection, especially, one characteristic point may be selected or two or more may be selected depending on the situation. In addition, characteristic of an object may be expressed as a line instead of a point. Through existing papers, we can confirm that fingerprint authentication is possible by applying geometric hashing which was materialized as software [10] and [11]. And we also confirmed that the result of experiment revealed the performance to be ‘0’ for False Acceptance Rate (FAR) and ‘0.08’ for False Reject Rate (FRR) in the case of using the 9th degree polynominal [11].

3. Parallel Processing of the Fuzzy Fingerprint Vault

Clancy and Uludag proposed the fuzzy fingerprint vault theory which can carry out fingerprint authentication using the fuzzy vault [8][9]. In case a fingerprint template is attacked, user's fingerprint information can be used illegally. To solve this problem, the fuzzy fingerprint vault system generates and inserts multiple chaff minutiae into the fingerprint template, as shown in **Fig. 4**.

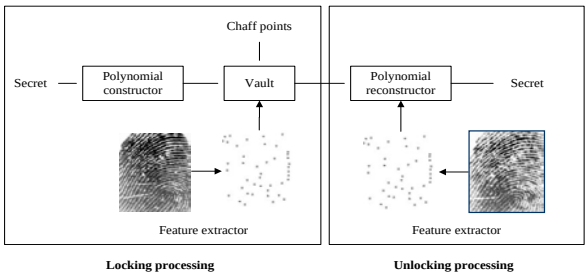
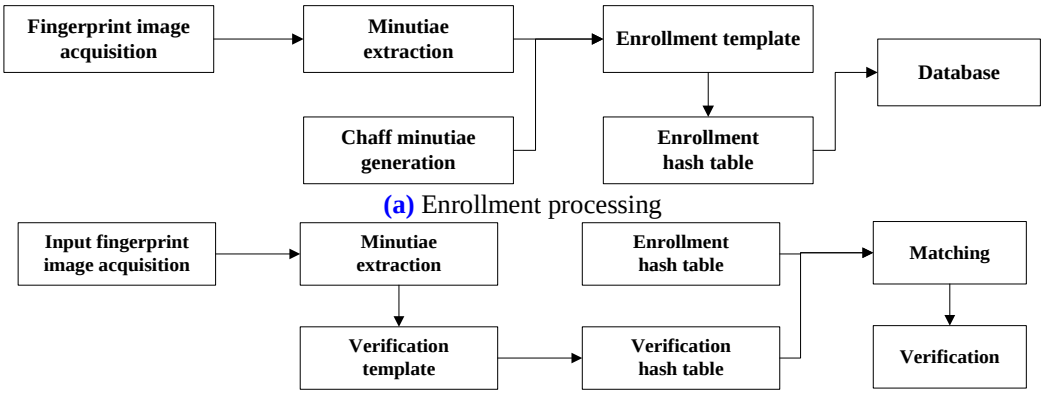


Fig. 4. Fuzzy fingerprint vault block diagram

The fingerprint template can carry out fingerprint authentication for the user in un-restored state like this. When a protected fingerprint template is stolen, the attacker cannot separate the real minutiae of the user from the fingerprint template which chaff minutiae has been added. Fuzzy fingerprint vault has a problem in fingerprint alignment. For fingerprint alignment, some methods, such as Fingerprint Orientation Model Based on 2D Fourier Expansion(FOMFE), has been proposed [20]. However, the singular point of fingerprint does not always exist in the image. Also, Fingerprint alignment is carried out by using structural information that is mutually formed by real minutiae. Fuzzy fingerprint vault includes hundreds of false information in the fingerprint information. It is difficult to differentiate real minutiae and added chaff minutiae from the fingerprint template. Therefore, there is difficulty in performing the alignment process because accurate structural information cannot be used. The existing studies have problems of not being realistic and not implementing an automated system because the alignment process which is the intermediate stage of fingerprint recognition is assumed to perform or is aligned manually [7][8][9]. In order to solve this problem, Chung et al. carried out the alignment automatically by applying the geometric hashing technique [10][11][12][13].



(a) Enrollment processing

(b) Verification processing

Fig. 5. Fuzzy fingerprint vault system

Geometric hashing technique is a method which alignment is performed automatically by selecting both the user's minutiae which consist of the fingerprint template and chaff minutiae as reference point. Enrollment processing of fuzzy fingerprint vault based on geometric hashing is as shown in **Fig. 5-(a)**. Enrollment table for which geometric characteristic of the minutiae is considered is generated by extracting the minutiae of the fingerprint and adding chaff minutiae. Enrollment template (Locking set) which chaff minutiae and user's real minutiae are included can be expressed as $L = \{ m_i \mid 1 \leq i \leq r \}$. In L, set for the user's real minutiae and chaff minutiae can be expressed $G = \{ m_i \mid 1 \leq i \leq n \}$, $C = \{ m_i \mid n+1 \leq i \leq r \}$ respectively. The enrollment template generates the enrollment hash table through geometric hashing technique. Select a reference point for all minutiae of G, the user's real minutiae, and C, the chaff minutiae. The hash table is generated by rotation and movement transformation of the remaining minutiae after excluding the selected reference point. A hash table consists of transformed minutiae generated by geometric transformation.

In the verification processing, minutiae information extraction step, table generation step, matching step and verification step (Candidate list generation step) are executed sequentially, as shown in **Fig. 5-(b)**. The minutiae information extraction step and table generation step are performed by identical process and order as the enrollment processing. The matching step is a

step which minutiae pair for which the position and angle match are found by using the transformed minutiae set as the basic unit. The measurement is possible by finding the number of transformed minutiae pairs that match between the set of transformed minutiae for the enrollment table stored in the database and transformed minutiae for the verification table. The set of minutiae made like this becomes the unlocking set U [12]. In the verification step, similarity is measured by comparing the set of all transformed minutiae for the enrollment table stored in the fingerprint database with respect to the set of all transformed minutiae in the recognition table. Candidate list is then found in high similarity order.

3.1 Hardware Architecture of the Fuzzy Fingerprint Vault

To implement the hardware system of the fuzzy fingerprint vault, the proposed architecture was performed by integrating software and hardware modules as shown in **Fig. 6**. The enrollment processing for the fuzzy fingerprint vault system consists of steps for real minutiae extraction, chaff minutiae generation, fingerprint template generation, fingerprint hash table generation and fingerprint database storage. The verification processing consists of steps for input fingerprint minutiae extraction, fingerprint hash table generation, matching and verification (Candidate list generation step).

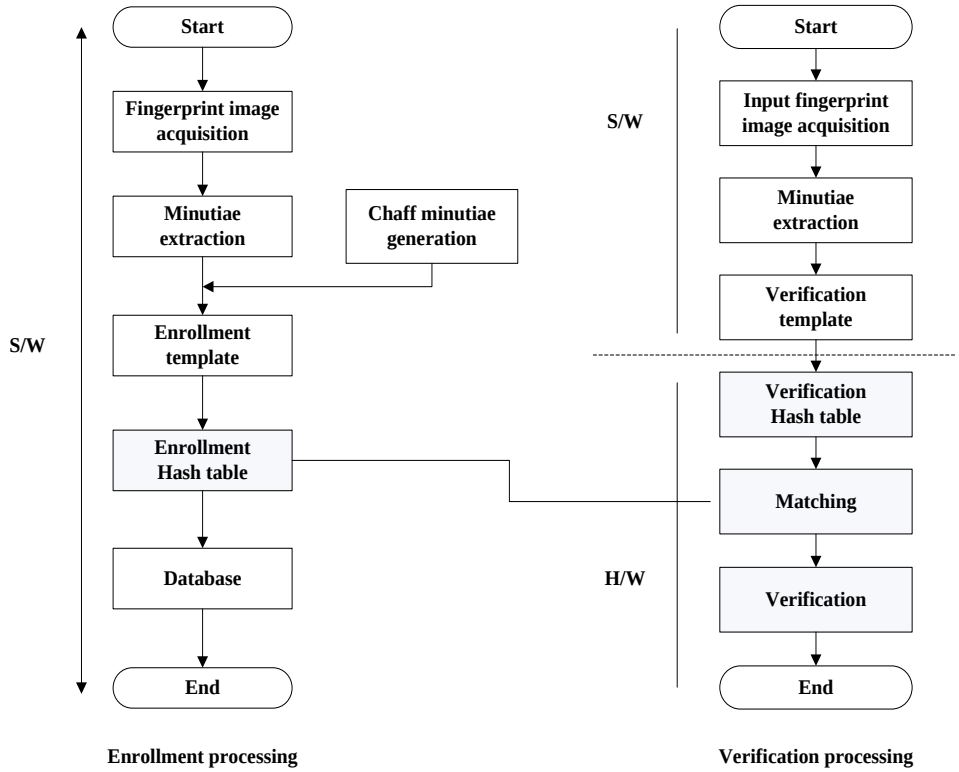


Fig. 6. Integrated implementation of S/W and H/W for the fuzzy fingerprint vault system

The matching step and verification step of the verification processing are performed in hardware. Since the amount of computation of the matching step for the verification processing increases as the number of chaff minutiae increases in the enrollment processing, it is advisable for the matching step and verification step to be performed in hardware. On the other hand, the enrollment processing and fingerprint hash table generation step in the

verification processing are performed in software [21].

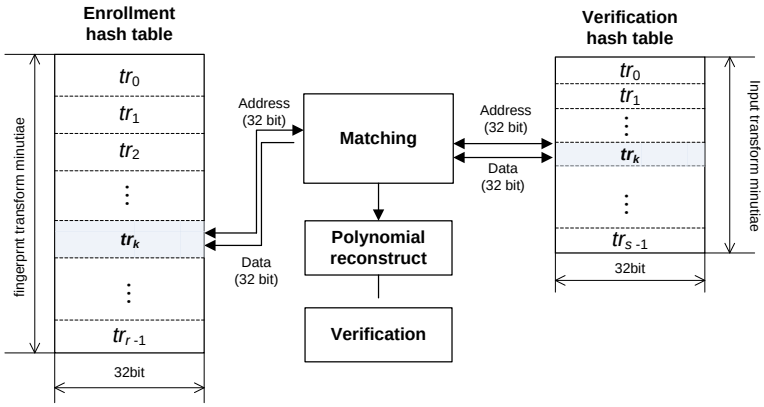


Fig. 7. Structural diagram of the proposed hardware module

The proposed hardware module consists of two memories for storing the enrollment hash table and verification hash table, matching module and verification module as shown in **Fig. 7**. Hash table for each is organized of transformed minutiae. The enrollment fingerprint transformed minutiae is a minutiae created through geometric transformation of user's real minutiae and chaff minutiae that was inserted to protect this in the enrollment processing. The verification fingerprint transformed minutiae is a minutiae created by geometric transformation of minutiae of the fingerprint for verification. The transformed minutiae consists of a total of 32bits consisting of 11bits each for the x axis and y axis, 9bits for the angle and 1bit for the type. The enrollment hash table can be expressed as $E = \{tr_i \mid 0 \le i \le r\text{-}1\}$ where r is the number of enrollment fingerprint templates. tr_i is the hash table which is created through geometric hashing after selecting among the fingerprint template as the reference point and consists of r-1 transformed minutiae. The verification fingerprint transformed minutiae can be expressed as $V = \{tr_j \mid 0 \le j \le s\text{-}1\}$ where s is the number of verification fingerprint templates. For the size of the memory to store the hash table, p, the number of enrollment fingerprint transformed minutiae for each, is $(r \times (r\text{-}1))$ and q, the number of verification fingerprint transformed minutiae, is $(s \times (s\text{-}1))$.

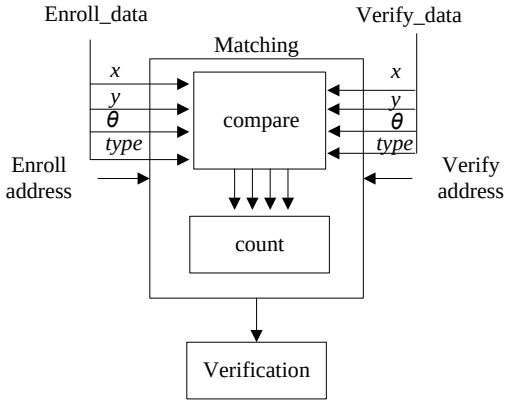


Fig. 8. Structural diagram of the proposed matching module and verification module

The architecture of the matching module and verification module is as shown in **Fig. 8**. The matching module consists of the Compare module and Count module. The Compare module is

the module that compares the enrollment fingerprint transformed minutiae and verification fingerprint transformed minutiae and the Count module calculates the number of corresponding transformed minutiae. The verification module is the module that aligns calculated similarity in high similarity order.

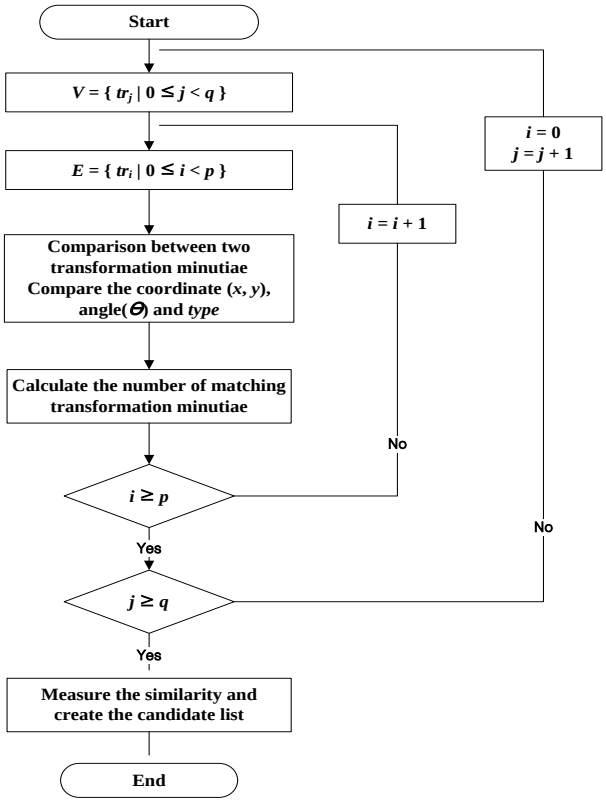


Fig. 9. Flow chart for the proposed hardware module

The flowchart for the alignment method of the proposed hardware architecture is as shown in **Fig. 9**. It assumes the number of enrollment fingerprint templates to be *r*, *tr* of the enrollment hash table *E* to be *r*, number of the verification fingerprint template to be *s* and *tr* of the verification hash table *V* to be *s*. First, tr_0 of the verification hash table and tr_0 of the enrollment hash table are input to the matching module. In the Compare module of the matching module, tr_0 of the verification hash table and tr_0 of the enrollment hash table are compared. After comparing the coordinate, angle and type of the transformed minutiae for the two *tr* that were input, whether or not they match is sent to the Count module. The Count module calculates the number of transformed minutiae that match. Then, tr_1 of the enrollment hash table is input to the matching module and compared with the tr_0 of the verification hash table. After comparison of all *tr* in the enrollment hash table with tr_0 of the verification hash table are completed using the same method, tr_1 of the verification hash table is input to the matching module. Up to tr_{s-1} of the verification hash table is compared by executing this repeatedly.

For example, let's assume that the number of enrollment fingerprint template is 219 (19 real minutiae and 200 chaff minutiae) and number of verification fingerprint template is 10. The number of *tr* for the enrollment hash table is 219 and the verification hash table consists of 10 *tr*. After tr_0 of verification hash table and tr_0 of enrollment hash table have been input into the

matching module and the two *tr* have been compared, the number of corresponding transformed minutiae is calculated. Then, enrollment hash table's tr_1 is input to the matching module and compared with tr_0 of verification hash table and the corresponding transformed minutiae is calculated. Using the same method, tr_{218} of the enrollment hash table is input to the matching module and compared with tr_0 of the verification hash table. Then, tr_9 of the verification hash table is input to the matching module and then tr_0 of the enrollment hash table is input again and compared. Up to tr_9 of the verification hash table is compared by executing this repeatedly.

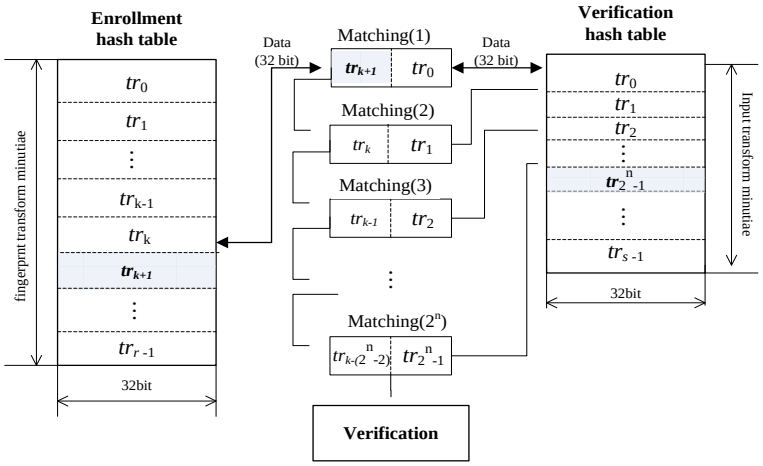


Fig. 10. Structural diagram of the proposed parallel processing technique hardware module

3.2 Parallel Processing Hardware Architecture of the Fuzzy Fingerprint Vault

In this paper, we propose a hardware architecture which parallel processing technique is applied to reduce the matching time. Separation of the software and hardware modules is identical to the previously proposed hardware architecture. The proposed parallel processing hardware module consists of two memories storing the enrollment hash table and verification hash table, matching module and verification module as shown in **Fig. 10**. For parallel processing, the number of matching modules used in the hardware module is 2^n $(1 \leq n \leq 6)$.

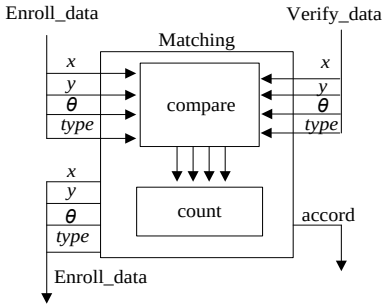


Fig. 11. Structural diagram for the hardware matching module with the proposed parallel processing technique

While the architecture of the matching module is similar to the previously proposed hardware architecture in **Fig. 11**, it is different because the input enrollment fingerprint transformed minutiae is output after the comparison. The architecture of the verification

module is identical to the previously proposed hardware architecture. When the number of the enrollment fingerprint templates is *r* and the number of the verification fingerprint templates is *s*, the enrollment hash table *E* consists of *r tr* and verification hash table *V* consists of *s tr*.

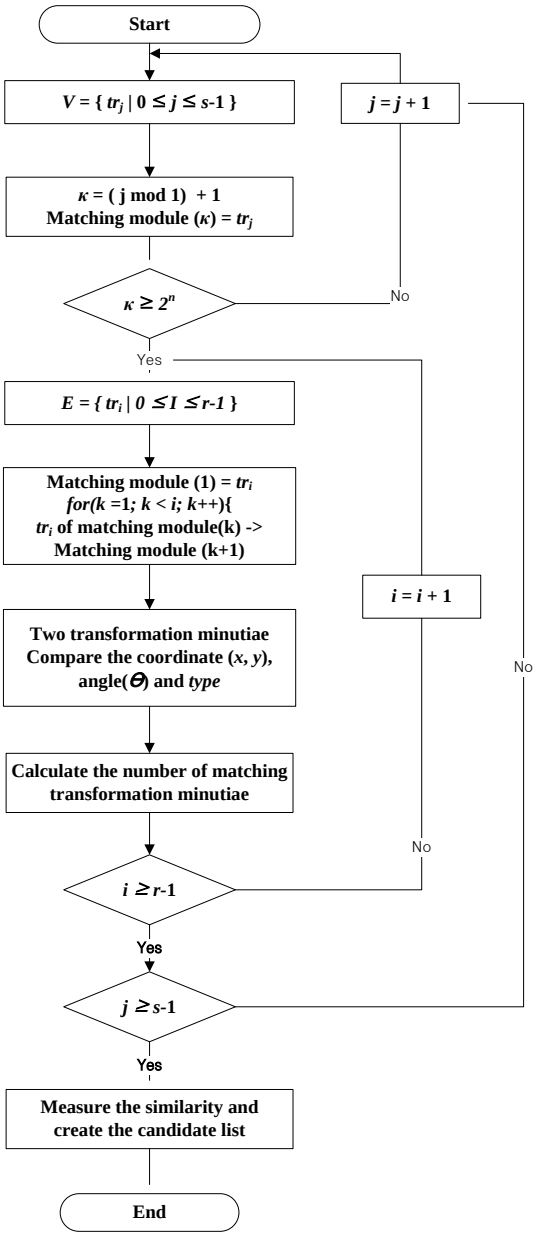


Fig. 12. Flow chart for the hardware module with the proposed parallel processing technique

For example, hardware architecture consisting of 2 matching modules is as follows. First, tr_0 and tr_1 of the verification hash table is input into matching module (1) and matching module (2) respectively and tr_0 of the enrollment hash table is input to matching module (1) (Matching module (1) and matching module (2) are displayed in **Fig. 10** as Matching (1) and Matching (2)). The transformed minutiae of the two *tr* (tr_0 for the verification hash table and tr_0 for the enrollment hash table) that were input from the Compare module of matching

module (1) are compared. The numbers of pairs for the corresponding transformed minutiae are calculated from the Count module. Then, tr_0 of the enrollment hash table is sent from matching module (1) to matching module (2) and tr_1 of the enrollment hash table is sent from the enrollment hash table to matching module (1). Matching module (1) compares $tr_1$of the enrollment hash table and tr_0 of the verification hash table and matching module (2) compares tr_0 of the enrollment hash table and tr_1 of the verification hash table simultaneously. After determining whether or not the transformed minutiae matches, *tr* of the enrollment hash table is sent to the next matching module. After executing until comparison for all *tr* of the enrollment hash table causes with tr_1 of the verification hash table in matching module (2) by using the identical method given above, tr_2 and tr_3 of the verification hash table are input to matching module (1) and matching module (2). The process above is carried out as tr_0 is sent to the matching module (1) again. This is repeated until tr_{s-1} of the verification hash table is input to matching module (2) and compared with $tr_{r-1}$of the enrollment hash table.

The architecture of the hardware module consisting of the following 4, 8, 16, 32 and 64 matching modules only differ in the number of matching modules and are executed with the identical process.

Flowchart for the hardware with the proposed parallel processing technique is as shown in **Fig. 12**. For example, let's assume that the number of the enrollment fingerprint templates is 213 (13 real minutiae and 200 chaff minutiae) and it is 12 for the verification fingerprint template. The enrollment hash table consists of 213 *tr* and verification hash table consists of 12 *tr*. First, tr_0 and tr_1 of the verification hash table is input into matching module (1) and matching module (2) respectively. Then tr_0 of the enrollment hash table is input into matching module (1) and compared with tr_0 of the verification hash table. After determining whether the two *tr* match, tr_0 of the enrollment hash table is sent from matching module (1) to matching module (2) and tr_1 of enrollment hash table is sent from the enrollment hash table to the matching module (1). This is carried out using an identical method until tr_{212} of the enrollment hash table is compared with $tr_1$of the verification hash table in matching module (2). Then, tr_2 and tr_3 of the verification hash table is input into matching module (1) and matching module (2). This is repeated until tr_{11} of the verification hash table is input into the matching module and compared with $tr_{212}$of the enrollment hash table.

4. Experimental Results and Analysis

The number of real minutiae that was used in the hardware architecture experiment for the fuzzy fingerprint vault system based geometric hashing proposed in this paper is between a maximum of 90 and minimum of 16. Fuzzy vault and geometric hashing methods used in this paper were materialized based on [10]. The experiment was performed by adding 100, 200, 300 and 400 chaff minutiae. Here the templates for registered fingerprint does not have any information which can distinguish real and chaff, and the templates, which cannot distinguish real and chaff as three characteristic points for real and chaff are mixed randomly, were created and use. The software module was realized by using C language in Visual C++ 6.0.

The major specifications of the PC are followed as below.

· Intel Core2 Duo E7300 2.67GHz
· SDRAM 2.00GB
· Windows Vista 32bit Operation System

To simulate the hardware architecture proposed in this paper, Spartan 3E start board was used. The development board contains a Xilinx XC3S500E FPGA. The hardware module was designed by using VHDL in Xilinx ISE 9.2. Hardware simulation was performed in Moledmsim XE 6.0.

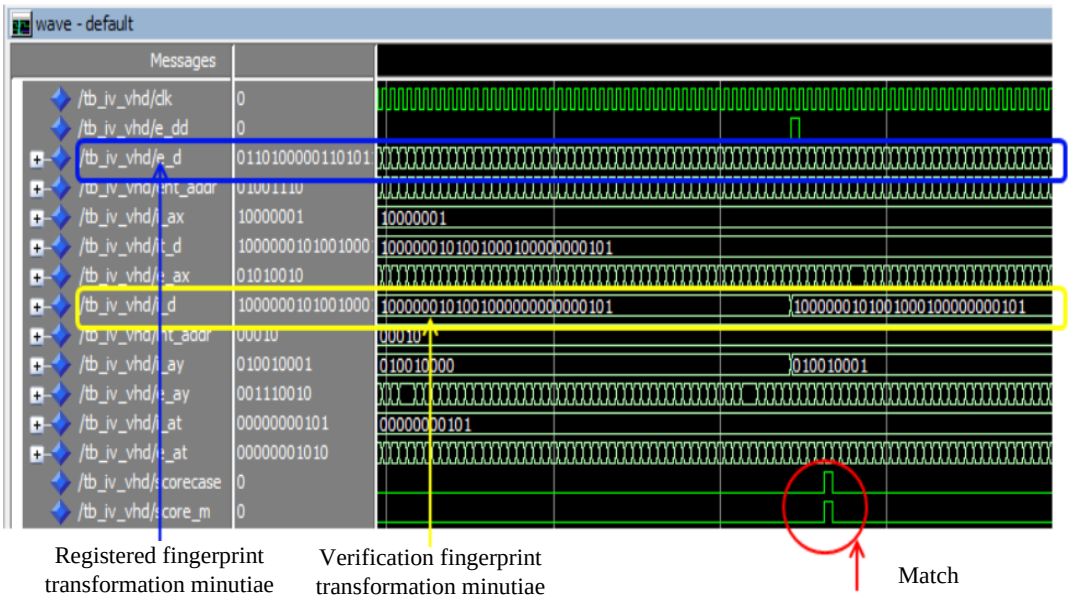


Fig. 13. Simulation result for the hardware matching module

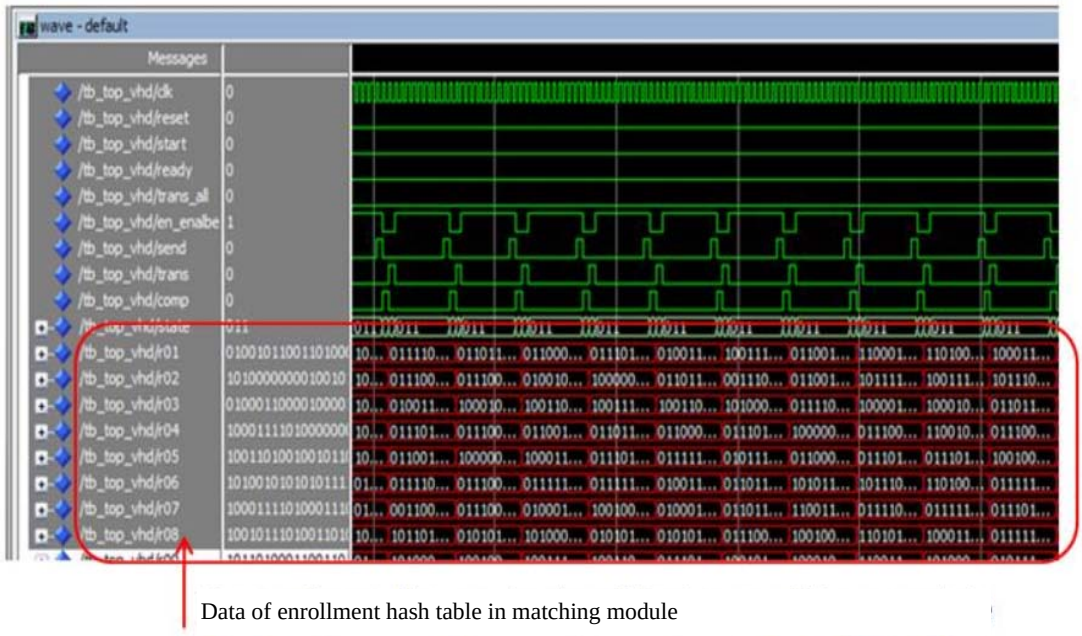


Fig. 14. Simulation result for the parallel processing hardware matching module (For 64 matching modules)

Fig. 13 shows the matching module simulation result for the proposed hardware architecture in the fuzzy fingerprint vault system. The experiment was performed by adding 36

real minutiae and 200 chaff minutiae. There were 34 verification fingerprint minutiae. Matching for the proposed hardware architecture compares the input fingerprint hash transformed minutiae and enrollment fingerprint hash transformed minutiae. It determines whether there is a match between the two transformed minutiae and measures the similarity by counting its number.

Fig. 14 shows the simulation result for the matching module of the hardware architecture (64 matching modules) which the proposed parallel processing technique was applied. The number of real minutiae and chaff minutiae are identical to the experiment that was executed previously. The red waveform in **Fig. 14** shows the transformed minutiae of the enrollment fingerprint hash table inside each matching module.

Table 1. Major resources when matching module is implemented in a Xilinx Spartan 3E FPGA

Number of matching modules for the proposed hardware architecture	Number of Slices	Number of Slices Flip Flop	Total Number of 4 input LUTs
1	419 out of 4,656 (9%)	161 out of 9,312 (2%)	668 out of 9,312 (7%)
2	496 out of 4,656 (10%)	246 out of 9,312 (2%)	762 out of 9,312 (8%)
4	822 out of 4,656 (17%)	618 out of 9,312 (6%)	951 out of 9,312 (10%)
8	1,046 out of 4,656(26%)	946 out of 9,312(10%)	1,931 out of 9,312(20%)
12	1,963 out of 4,656(42%)	1,161 out of 9,312(12%)	3,619 out of 9,312(38%)
32	2,793 out of 4,656(59%)	2,015 out of 9,312(21%)	4,962 out of 9,312(53%)
64	3,048 out of 4,656(65%)	2,856 out of 9,312(30%)	6,081 out of 9,312(65%)

Table 1 shows the major resources that were used when the hardware architecture for the proposed fuzzy fingerprint vault and hardware architecture which the parallel processing technique was applied were implemented in the development board. Since the parallel processing hardware architecture has matching modules whose number equals that of the verification fingerprint transformed minutiae when there is one matching module in the proposed hardware architecture, the amount of resources that are used is large. It can be seen that the amount of hardware is about 56% higher for the number of slices, about 28% higher for the number of slice flip flops and 58% higher for the total number of 4 input LUTs in the parallel processing hardware architecture.

Table 2. Required number of cycles according to the number of chaff minutiae

Chaff minutiae / Module	Number of chaff minutiae			
	100	200	300	400
Software	30,128,840	56,376,516	75,087,761	110,009,410
1 matching module	10,023,288	12,027,946	18,763,596	26,461,481
2 matching module	8,173,713	9,808,456	15,301,191	21,578,603
4 matching module	4,086,907	4,904,288	7,650,689	10,789,434
8 matching module	2,043,453	2,452,144	3,825,345	5,394,717
16 matching module	1,021,714	1,226,057	1,912,679	2,697,325
32 matching module	510,857	613,028	956,324	1,348,662
64 matching module	480,807	576,968	900,070	1,239,330

Execution time of the hardware module for the fuzzy fingerprint vault system was measured with number of chaff minutiae equal to 100, 200, 300 and 400. There is correlation between the number of chaff minutiae, execution time and security. While execution time is increased as the number of chaff minutiae becomes larger, security is strengthened. As shown in **Table 2**, real time processing is possible even when the number of chaff minutiae is increased to

improve security in the proposed hardware architecture. **Fig. 15** shows the time calculated by applying the 50MHz clock frequency of development board based on the Table 2. As for the proposed hardware architecture, we confirmed that the real-time processing is possible even after increasing the number of chaff minutiae for the security improvement.

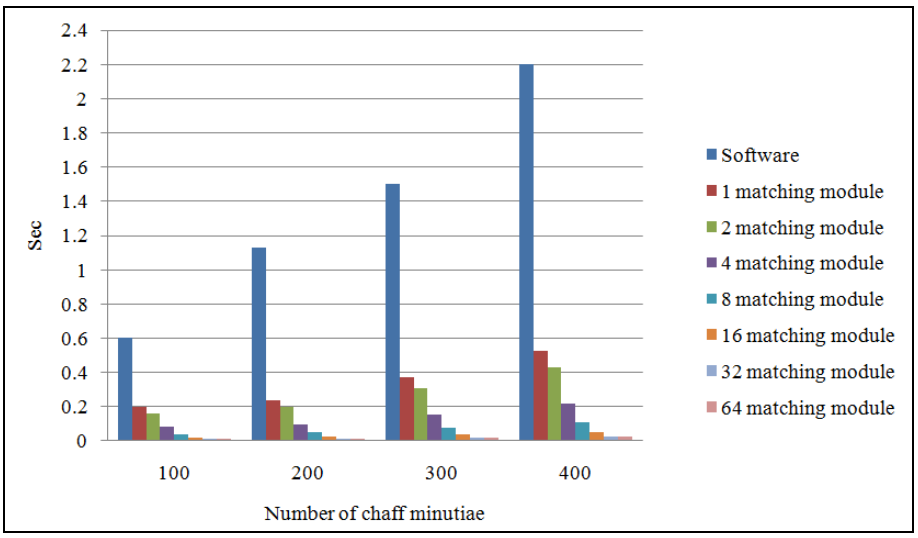


Fig. 15. Execution time according to the number of chaff minutiae

5. Conclusions

While a user authentication system using fingerprint information provides convenience and strong security at the same time, serious problems may cause if the fingerprint information is used illegally or leaked. Recently, study on fuzzy fingerprint vault which fuzzy vault theory is applied has been performed to protect the fingerprint template. In this paper, we proposed hardware architecture for a geometric hashing based fuzzy fingerprint vault system. The matching module of the proposed hardware architecture was performed so that all transformed minutiae in the enrollment fingerprint hash table are matched with each transformed minutiae in the verification fingerprint hash table. In addition, the hardware architecture of the matching system which parallel processing technique was applied for high speed processing of the system organizes matching modules in number equal to the number of transformed minutiae in the input fingerprint hash table and matches them simultaneously. Execution time of the proposed system was 0.24 second for 36 real minutiae and 200 chaff minutiae and 0.53 second for 400 chaff minutiae. In addition, execution time for hardware architecture with 64 matching modules which parallel processing technique was applied was 0.01 second and 0.03 second respectively for the same condition. Based on the experimental result, it was verified that real-time fingerprint authentication is possible by using the hardware architecture of the proposed fuzzy fingerprint vault system and high speed processing is possible by applying parallel processing technique. In the future, study will be carried out for a restoration method which polynomial for fuzzy fingerprint vault is applied and for a method to implement this in hardware. And in order to apply fuzzy vault in the restrictive environment like Smart Card, the author plans to study on the method of reducing the magnitude and complexity of data of the template.

References

[1] W. Stallings, *Cryptography and Network Security*, Pearson Ed. Inc., 2003. Article (CrossRef Link)

[2] S. Prabhakar, S. Pankanti and A. Jain, "Biometric recognition: security and privacy concerns," in *Proc. of IEEE Security and Privacy*, vol. 1, no. 2, pp. 33-42, 2003. Article (CrossRef Link)

[3] K. Y. Shin, B. J. Kang and K. R. Park, "Super-resolution iris image restoration using single image for iris recognition," in *Proc. of Transactions on Internet and Information Systems*, vol. 4, no. 2, pp. 117-137, 2010.

[4] D. Maltoni, D. Maio, A.K. Jain and S.Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003. Article (CrossRef Link)

[5] R. Bolle, J. Connell and N. Ratha, "Biometric perils and patches," in *Proc. of the Pattern Recognition*, vol. 35, pp. 2727-2738, 2002. Article (CrossRef Link)

[6] B. Schneier, "The uses and abuses of biometrics," in *Proc. of the Communications of the ACM*, vol. 42, no. 8, pp. 136, 1999. Article (CrossRef Link)

[7] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. of Symp. on Information Theory*, pp. 408, 2002. Article (CrossRef Link)

[8] T. Clancy, N. Kiyavash and D. Lin, "Secure smartcard-based fingerprint authentication," in *Proc. of ACM SIGMM Multim., Biom. Met. & App.*, pp. 45-52, 2003. Article (CrossRef Link)

[9] U. Uludag, S. Pankanti and A. Jain, "Fuzzy vault for fingerprints," *LNCS* 3546, pp. 310-319, 2005. Article (CrossRef Link)

[10] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim and D. Ahn, "Automatic alignment of fingerprint features for fuzzy fingerprint vault," *LNCS* 3822, pp. 358-369, 2005. Article (CrossRef Link)

[11] D. Moon, S. Lee, Y. Chung, S. B. Pan and K. Moon, "Implementation of automatic fuzzy fingerprint vault," in *Proc. of the Machine Learning and Cybernetics*, vol. 7, pp. 3781-3786, 2008. Article (CrossRef Link)

[12] D. Moon, S. Lee, S. Jung, Y. Chung, M. Park and O. Yi, "Fingerprint template protection using fuzzy vault," *LNCS* 4707, pp. 1141-1151, 2007. Article (CrossRef Link)

[13] Y. Chung, D. Moon, S. Pan, M. Kim and K. Kim, "A hardware implementation of fingerprint verification for secure biometric authentication," *LNCS* 3212, pp. 770-777, 2004.

[14] H. Wolfson and I. Rigoutsos, "Geometric hashing: an overview," in *Proc. of the IEEE Computational Science and Engineering*, vol. 4, no. 4, pp. 10-21, 1997. Article (CrossRef Link)

[15] W. Scheirer and T. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proc. of IEEE Biometrics Research Symposium at the National Biometrics Consortium Conf.*, pp. 1-16, 2007. Article (CrossRef Link)

[16] K. Nandakumar, A. Nagar and A. K. Jain, "Hardening fingerprint fuzzy vault using password," *LNCS* 4642, pp. 927-937, 2007. Article (CrossRef Link)

[17] P. Li, x. Yang, K. Cao, P. shi and J. Tian, "Security-enhanced fuzzy fingerprint vault based on minutiae's local ridge information," *LNCS* 5558, pp. 930-939, 2009. Article (CrossRef Link)

[18] A. Juels and M, Wattenberg, "A fuzzy commitment scheme," in *Proc. of ACM Conf. on Computer and Comm. Security*, pp. 28-36, 1999. Article (CrossRef Link)

[19] Y. Lamdan and H. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," in *Proc. of 2nd International Conf. Computer Vision*, pp. 238-249, 1998. Article (CrossRef Link)

[20] Y. Wang, J. Hu and D. Philips, "A fingerprint orientation model based on 2D Fourier Expansion (FOMFE) and its application to singular-point detection and fingerprint Indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 573-585, 2007. Article (CrossRef Link)

[21] S. Lim, S. Chae, D. Moon, Y. Chung, N. Lee and S. B. Pan "VLSI architecture for the fuzzy fingerprint vault with automatic alignment module," in *Proc. International Conf. FGCN*, vol. 56, pp. 470-476, 2009. Article (CrossRef Link)

Seung-Hoon Chae received the B.S. degree in electrical engineering in 2007 from Chosun University, Gwangju, Korea. He received the M.S. degrees in information and communication engineering in 2009 from Chosun University, Gwangju, Korea. He is currently working toward the Ph.D. degree. His research interests include design of embedded systems and biometrics.

Sung Jin Lim received the B.S. degree in control, instrumentation, and robot engineering in 2008 from Chosun University, Gwangju, Korea. He received the M.S. degrees in information and communication engineering in 2010 from Chosun University, Gwangju, Korea. His research interests include design of embedded systems and biometrics.

Sang-Hyun Bae received his B.S. and M.S. degrees in Electrical and Electronic Engineering from Chosun University, Korea in 1982 and 1984, respectively. He received his Ph.D. degree from the Tokyo Metropolitan University, Japan in 1988. He was a visiting professor at JAIST in 1996. He is now professor at Chosun University. He is a director of NRF(National Research Foundation of Korea). His research interests include digital contents, neural network, large know bases, and u-Health system.

Yongwha Chung received his B.S. and M.S. degrees from Hanyang University, Korea in 1984 and 1986, respectively. He received his Ph.D. degree from the University of Southern California, USA in 1997. He was a team leader at Biometric Technology Research Team of ETRI from 1986 to 2003. He is now professor at Korea University. His research interests include biometrics, security, and performance optimization.

Sung Bum Pan received the B.S., M.S., and Ph.D. degrees in Electronics Engineering from Sogang University, Korea, in 1991, 1995, and 1999, respectively. He was a team leader at Biometric Technology Research Team of ETRI from 1999 to 2005. He is now professor at Chosun University. His current research interests are biometrics, security, and VLSI architectures for real-time image processing.