Enhancement of an Automatic Fingerprint Identification System Using a Genetic Algorithm and Genetic Programming

Wannasak Wetcharaporn[1], Nachol Chaiyaratana[1] and Sanpachai Huvanandana[2]

[1] Research and Development Center for Intelligent Systems, King Mongkut's Institute of Technology North Bangkok
Bangkok, Thailand 10800, w_wannasak@hotmail.com, nchl@kmitnb.ac.th
[2] Department of Electrical Engineering, Chulachomklao Royal Military Academy, Nakhonnayok, Thailand 26001
shuvanan@crma.ac.th

ABSTRACT

This paper presents the use of a genetic algorithm and genetic programming for the enhancement of an automatic fingerprint identification system (AFIS). The recognition engine within the original system functions by transforming the input fingerprint into a feature vector or fingercode using a Gabor filter bank and attempting to create the best match between the input fingercode and the database fingercodes. A decision to either accept or reject the input fingerprint is then carried out based upon whether the norm of the difference between the input fingercode and the best-matching database fingercode is within the threshold or not. The efficacy of the system is in general determined from the combined true acceptance and true rejection rates. In this investigation, a genetic algorithm is applied during the pruning of the fingercode while the search by genetic programming is executed for the purpose of creating a mathematical function that can be used as an alternative to the norm operator. The results indicate that with the use of both genetic algorithm and genetic programming the system performance has improved significantly.

Keywords: Automatic Fingerprint Identification System, Genetic Algorithm, Genetic Programming

1. INTRODUCTION

Biometrics is an automated technique for identifying an individual based upon his or her physical or behavioural characteristics. The physical characteristics that are generally utilised as biometrics cover faces, retinae, irises, fingerprints and hand geometry while the behavioural characteristics that can be used include handwritten signatures and voiceprints. Among various biometrics, fingerprint-based identification is the most mature and proven technique. A fingerprint is made up from patterns of ridges and furrows on the surface of a finger [1]. The uniqueness of a fingerprint can be explained via (a) the overall pattern of ridges and furrows and (b) the local ridge anomalies called minutiae points such as a ridge bifurcation and a ridge ending. As fingerprint sensors are nowadays getting smaller and cheaper, automatic fingerprint identification systems (AFISs) have become popular alternatives or complements to traditional identification methods. Examples of applications that have adopted an AFIS are ranging from security control with a relatively small database to criminal identification with a large database.

Research in the area of fingerprint-based identification can be divided into two categories: fingerprint classification and fingerprint recognition. The purpose of classification is to cluster a database of fingerprints into sub-categories where the sub-categories are in general defined according to a Henry system [2]. Several techniques including syntactic approaches [3], structural approaches [4], neural network approaches [5] and statistical approaches [6] have been successfully used in fingerprint classification. In contrast, the purpose of recognition is to match the fingerprint of interest to the identity of an individual. A fingerprint recognition system is widely used in security-related applications including personnel identification and access control. For the purpose of access control, the goal of recognition is (a) to identify correctly a system user from the input fingerprint and grant him or her an appropriate access and (b) to reject non-users or intruders. Various techniques including minutiae-based approaches [7] and texture-based approaches [8] have been applied to fingerprint recognition. Among these techniques, the approach involving the transformation of a fingerprint into a fingercode [8] has received much attention in recent years. In brief, a fingerprint is transformed via a Gabor filter-based algorithm where the resulting feature vector or fingercode is a fixed length string that is capable of capturing both local and global details in a fingerprint. The fingerprint recognition is then achieved by matching the fingercode interested with that in the database via a vector distance measurement. Since the fingerprint is now represented by a unique fixed length vector and the matching mechanism is carried out through a vector operation, this approach has proven to be reliable, fast and requiring a small database storage.

Although a number of impressive results have been reported in Jain et al. [8], the recognition capability of the fingercode system can be further enhanced. One possible approach to improve the system is to modify the fingercode using a feature pruning technique. In most pattern recognition applications, the original feature vector is often found to be containing a number of redundant features. Once these features are removed, the recognition efficacy is in general maintained or improved in some cases. The most direct advantage for pruning the fingercode is the reduction in the database storage requirement. The candidate technique for pruning the fingercode is a genetic algorithm [9] where the decision variables indicate the presence and absence of features while the optimisation objective is the recognition efficacy. In addition to the feature pruning approach, the recognition system can also be improved by modifying the fingercode matching mechanism. In the original work by Jain et al. [8], a vector distance between the input fingercode and the database fingercode is used to provide the degree of matching. As a result, the distance value from each feature will contribute equally to the judgment on how well two fingercodes match one another. In this investigation, the mathematical structure for obtaining the distance and the level of contribution from each feature will be manipulated and explored using a genetic programming technique [10]. This part of the investigation is carried out in order to further increase the recognition capability of the system from that achieved after the feature pruning.

The organisation of this paper is as follows. In section 2, a brief explanation on the original fingercode system will be given. This also includes the description of the fingercode, which is the feature vector, and the matching mechanism. The application of the genetic algorithm on the feature pruning and the results will be discussed in section 3. Following that, the use of the genetic programming in matching mechanism modification and the results will be given in section 4.

2. FINGERCODE SYSTEM

The fingercode system developed by Jain et al. [8] consists of two major stages: filter-based feature extraction and fingercode matching stages. These two components are explained as follows.

2.1 Filter-Based Feature Extraction

There are three main steps in the feature extraction process described in Jain et al. [8]: (a) determination of a reference frame from the fingerprint image, (b) filtering the image using a Gabor filter bank and (c) computation of the standard deviation of pixel values in sectors around the reference point in the filtered image to obtain a feature vector or fingercode. Firstly, the point of maximum curvature of ridges in the fingerprint image is initially located as the reference point. The reference axis is then defined as the axis of local symmetry at the reference point. Next, the region of interest is identified; the region is composed of n concentric bands around the reference point where each band is segmented into k sectors. In this paper, eight concentric bands are used and there are 16 sectors in each band. Thus there are a total of $16 \times 8 = 128$ sectors in the region of interest as shown in Figure 1. The region of interest is filtered using a Gabor filter [11] where the standard deviation of filtered pixels within each sector is subsequently used as a feature in the fingercode. With the use of eight Gabor filters per one fingerprint image, the total number of features in a fingercode for this paper is $8 \times 128 = 1{,}024$.

(a) (b)

Fig.1: *(a) The Reference Axis (b)The Reference Point (×), and the Region of Interest, which Consists of 128 Sectors*

2.2 Fingercode Matching

After a feature vector or fingercode has been extracted from the input fingerprint, an attempt to identify the best match between the fingercode obtained and that in the database is carried out. The best-matching fingercode from the database will be the one where a norm of the distance between itself and the input fingercode is minimal. In this paper, a 1-norm is used in the distance measurement. It is noted that other norms such as a Euclidean norm and an infinite-norm can also be used. The fingerprint recognition can then be carried out via comparing the best-matching norm with a threshold. If the norm is less than or equal to the threshold, the recognition will identify the input fingerprint as being a part of the database and hence belongs to one of the system users. On the other hand, if the norm exceeds the threshold, the system will reject the input fingerprint and decide that the fingerprint belongs to an intruder.

The efficacy of the fingercode system can be determined from the correctness of the system output after the matching procedure. False output from the system can generally be divided into two categories: a false acceptance and a false rejection. A false acceptance refers to the situation when the system identifies an input fingerprint as belonging to one of the users while in fact the fingerprint belongs to either another user or an intruder. In contrast, a false rejection refers to the case where the system falsely identifies an input fingerprint as belonging to an intruder while the fingerprint actually comes from one of the users. Hence, the system efficacy can be expressed in terms of the combined true acceptance and true rejection rates, a false acceptance rate (FAR) and a false rejection rate (FRR). In the following two sections, the improvement of the fingercode system by means of genetic algorithm and genetic programming searches will be given.

3. FEATURE PRUNING USING A GENETIC ALGORITHM AND DISCUSSIONS

An attempt on reducing the number of features in a fingercode using a genetic algorithm is made. The decision variables for the optimisation cover the presence and absence of features in the fingercode. The decision variables can thus be represented by a binary chromosome where '1' represents the presence of a feature whilst '0' signifies the absence of a feature. In this investigation, there are 1,024 features in the fingercode. As a result, the chromosome length is also equal to 1,024 bits. Since the size of the reduced feature vector can vary during the optimisation process, the value of threshold required for the decision made by the recognition system has to also be modified accordingly. In this paper, a 1-norm is used during the feature matching procedure. The threshold can thus be set such that it is linearly proportional to the number of remaining features in the fingercode after pruning. After the matching between all input fingercodes and the database fingercodes, and the acceptance/rejection decision has been made, the fitness value of each chromosome can be calculated from the combined true acceptance and true rejection rates expressed in percent. In this investigation, 400 fingerprint images are collected from 40 individuals where each individual contributes ten fingerprints. During a genetic algorithm run, 300 fingerprints from 30 individuals are retained within the user database while the other 100 fingerprints from the remaining ten individuals are used as fingerprints from intruders. All 400 fingerprints are transformed into fingercodes using the feature extraction procedure where the decisions to accept or reject the input fingercodes are subsequently made. It is noted that the fingerprint database remains unchanged throughout the genetic algorithm run, which in this case is repeated ten times with different initial populations. The parameter setting for the genetic algorithm is summarised in Table 1. After all ten algorithm runs are completed, reduced fingercodes as represented by the best individuals from all runs, and three additional chromosomes, which are resulted from applying an AND function, an OR function and a majority vote rule to aligned bits of all ten best individuals are then tested or validated. The fingerprint databases used for validation also comprise of fingerprints taken from the original 400 fingerprints. However, all except one fingerprint sets for validation would be different from that used during the genetic algorithm runs. Nine newly created databases and the original database are utilised during the validation where the results are illustrated in Table 2.

From Table 2, it can be clearly seen that the use of a pruned or reduced fingercode leads to an improvement in recognition performance over the use of a full fingercode for at least 7% in overall. The highest improvement comes from the case of reduced fingercode obtained after using a majority vote rule where detailed results indicate that there are improvements in both fault acceptance and fault rejection rates. On the other hand, the reduced fingercodes that have the worst performance are the ones resulted from the use of AND and OR functions. These results can be interpreted as follows. With the application of a majority vote rule in deciding whether a feature should be maintained or removed from the fingercode, the effect of uncertainties due to the stochastic search nature of genetic algorithms on the overall optimisation result would be minimised. During each genetic algorithm run, the search is conducted in a manner that maximises the recognition efficacy. Since the search is a stochastic one and there may be more than one globally optimal reduced fingercode, the use of a majority vote rule would help maintaining necessary features detected in most or all runs while at the same time eliminating possible redundant features. This reason is supported by the results where AND and OR functions are used, which indicate that there is no significant gain in recognition performance over the use of a reduced fingercode obtained from a typical genetic algorithm run. With the application of an OR function, the resulting fingercode would contain both necessary features and some redundant features while with the use of an AND function, some crucial features may be left out since they are not present in all best individuals. These two phenomena would have caused a reduction in the recognition performance.

Table 1: *Parameter Setting for the Genetic Algorithm*

Parameter	Setting and Value
Chromosome representation	Binary chromosome
Chromosome length	1,024
Fitness scaling method	Linear scaling
Selection method	Stochastic universal sampling
Crossover method	Uniform crossover ($p_c = 0.8$)
Mutation method	Bit-flip mutation ($p_m = 0.1$)
Population size	100
Number of generations	1,000
Number of repeated runs	10

Table 2: *Validation Results of the Fingercode System with and without Feature Pruning. The First Validation Set is also Used during All Repeated Runs of the Genetic Algorithm.*

Validate Set	Recognition Efficacy (%)				
	Original Finger-code (1,024 Feat.)	Example of a Reduced Finger-code (419 Feat.)	Finger-code from an AND Func. (384 Feat.)	Finger-code from an OR Func. (521 Feat.)	Finger-code from a Majority Vote (492 Feat.)
1	88.75	96.00	95.50	95.00	96.25
2	85.25	95.00	94.50	94.25	95.50
3	83.25	91.50	91.25	91.00	92.00
4	83.00	91.00	90.50	90.50	91.00
5	83.25	91.25	91.25	90.25	92.50
6	82.25	90.50	90.25	89.00	91.25
7	76.50	85.00	84.75	83.50	85.75
8	76.75	87.00	86.75	85.50	88.00
9	79.50	87.75	87.75	86.25	88.75
10	85.75	92.00	92.00	90.25	92.50
Average	82.43	90.70	90.45	89.55	91.35

4. MODIFICATION OF THE MATCHING MECHANISM USING GENETIC PROGRAMMING AND DISCUSSIONS

In the original work by Jain et al. [8] and the investigation so far, the decision to accept or reject input fingercode is based on whether the best-matching norm is within the threshold or not. In this section, the calculation of 1-norm will be replaced by the mathematical function or operation evolved by genetic programming (GP). Nonetheless, the output from the evolved function will still be compared with the threshold during the decision-making procedure. Since the use of a reduced fingercode generated by a majority vote rule has proven to produce the current best result, all features from this reduced fingercode will be used as a part of terminal set. The terminal set is thus made up from preset constant values and the absolute differences between the input features and the corresponding features from a reduced fingercode in the database. It is noted that the best-matching fingercode in the database is the one that the GP-evolved function returns the minimum value. The parameter setting for the genetic programming is summarised in Table 3.

Table 3: *Parameter Setting for the Genetic Programming*

Parameter	Setting and Value
Tree initialisation method	Grow method
Maximum tree depth	10
Terminal set	{Constants: 0.25, 0.50, 0.75, 1.25, 1.50, 1.75, 2.00 and absolute differences between input and database features}
Function set	{+, -}
Fitness scaling method	Linear scaling
Selection method	Stochastic universal sampling
Crossover probability (p_c)	0.8
Mutation probability (p_m)	0.1
Population size	100
Number of elitist individuals	1
Number of generations	2,000
Number of repeated runs	10

Table 4: *Validation Results of the Reduced-Feature Fingercode System with the Use of 1-Norm and GP-Generated Function during Matching*

Validate Set	Matching via 1-Norm			Matching via GP-Generated Function		
	Eff. (%)	FAR (%)	FRR (%)	Eff. (%)	FAR (%)	FRR (%)
1	96.25	0.75	3.00	98.00	0.50	1.50
2	95.50	0.75	3.75	97.00	0.50	2.50
3	92.00	0.25	7.75	93.00	0.25	6.75
4	91.00	1.00	8.00	91.75	1.00	7.25
5	92.50	0.00	7.50	92.50	0.25	7.25
6	91.25	0.50	8.25	92.25	0.25	7.50
7	85.75	5.50	8.75	88.25	3.75	8.00
8	88.00	2.75	9.25	91.25	1.00	7.75
9	88.75	2.50	8.75	91.75	1.25	7.00
10	92.50	2.50	5.00	95.00	2.25	2.75
Average	91.35	1.65	7.00	93.08	1.10	5.82

Similar to the approach presented in the previous section, ten databases are also used during the validation where the results are displayed in Table 4. The genetic programming results are produced using the best individual among all ten runs. From Table 4, it can be clearly seen that the replacement of 1-norm by the GP-generated function leads to a further improvement in terms of the recognition efficacy, fault acceptance rate and fault rejection rate from that achieved earlier. This also implies that the use of mathematical functions other than a norm function may be more suitable to the fingercode system. It is noticeable that the system performance is highest in the case of the first validation set, where it is also the data set used during the evolution of a matching function using genetic programming. This in general is true in most pattern recognition applications.

5. REFERENCES

[1] F. Galton, *Finger Prints*, Macmillan, London, UK, 1892.

[2] E. R. Henry, *Classification and Uses of Finger Prints*, HM Stationary Office, London, UK, 1905.

[3] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson, "Evaluation of Pattern Classifiers for Fingerprint and OCR Applications," *Pattern Recognition*, Vol. 27, No. 4, pp. 485-501, 1994.

[4] K. Karu and A. K. Jain, "Fingerprint Classification," *Pattern Recognition*, Vol. 29, No. 3, pp. 389-404, 1996.

[5] A. K. Jain, S. Prabhakar and L. Hong, "A Multichannel Approach to Fingerprint Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, pp. 348-359, 1999.

[6] L. Coetzee and E. C. Botha, "Fingerprint Recognition in Low Quality Images," *Pattern Recognition*, Vol. 26, No. 10, pp. 1441-1460, 1993.

[7] A. Farina, Z. M. Kovács-Vajna and A. Leone, "Fingerprint Minutiae Extraction from Skeletonized Binary Images," *Pattern Recognition*, Vol. 32, No. 5, pp. 877-889, 1999.

[8] A. K. Jain, S. Prabhakar, L. Hong and S. Pankanti, "Filterbank-Based Fingerprint Matching," *IEEE Transactions on Image Processing*, Vol. 9, No. 5, pp. 846-859, 2000.

[9] D. E. Goldberg, *Genetic Algorithms: In Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[10] J. R. Koza, *Genetic Programming: On the Programming by Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

[11] J. G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1148-1161, 1993.