Finger Print Matching and Template Security Using Fuzzy vault.

Ruturaj M. Dhekane
Multimodal Biometric Systems
Indian Institute Of Technology, Kanpur

Abstract—Template Security is an important issue in biometric systems because unlike passwords, stolen biometric templates cannot be revoked. A fully automatic implementation of a fingerprint-based fuzzy vault was implemented, that secures minutiae templates. This system secures knowledge and also gives higher GAR and low FAR rates. To support the fuzzy vault framework, a minutiae extraction algorithm and matching technique was designed. Four matching techniques were tested and analyzed and best was chosen for vault implementation.

Index Terms—Finger Print, Fuzzy Vault, Matching Algorithms

I. INTRODUCTION

Security is of prime concern in most establishments. Biometric systems come close to disallowing false person in accessing resources. Biometric template is stored in a system that enrolls allowed candidates. However this system is prone to hijack, and biometric templates can be stolen. Another issue is in the use of smart cards on which Biometric templates are stored. For a secure communication between the verification device and the card, a secret key needs to be generated. Management of these keys is a difficult job, and hence generation of key at enrolling of candidate, followed by deriving, the key using the biometric samples.

A Fuzzy vault is a cryptographic construct that is designed to work with biometric features presented as unordered set. The security of the fuzzy vault scheme is the infeasibility of the polynomial reconstruction problem. The fuzzy vault scheme adds random minutiae points to the template. These points are removed when a genuine user provides his biometric for verification, since his biometric matches with correct points in the template. These matched points are used to reconstruct a polynomial whose roots are the minutiae points to regenerate a vector of coefficients which is the knowledge being secured.

To test the system a minutiae extraction and matching algorithm was implemented. Four matching techniques were used to analyze their efficiency. First a system based on signal processing technique and matching of signals by cross correlation. The combinatorial algorithm for matching called the Hungarian algorithm was used to find maximum matching. The dynamic programming based edit distance method was used. It did not give good results as expected. Finally the neighborhood minutiae points were also stored and points were matched if their neighboring points also matched. This gave the best results and was used through the project as the matching algorithm.

II. RELATED WORK

Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify an individual and verify their identity.

Most work in this field started in 1990’s with Jain Et. Al publishing papers on finger print recognition. There have been lot of advanced in fingerprint recognition techniques and today the use to markov chains and orientation detection using singular extraction. All the related work reference papers that were used have been referred in the reference section.

III. FINGER PRINT EXTRACTION

Fingerprints are today the most widely used biometric features for personal identification. Most automatic systems for fingerprint comparison are based on minutiae matching. Minutiae characteristics are local discontinuities in the fingerprint pattern which represent terminations and bifurcations. A ridge termination is defined as the point where a ridge ends abruptly. A ridge bifurcation is defined as the point where ridge forks or diverges into branch ridges.

Reliable automatic extracting of minutiae is a critical step in fingerprint classification. The ridge structures in fingerprint images are not always well defined, and therefore, an enhancement algorithm, which can improve the clarity of the ridge structures, is necessary.

A. Finger Print Enhancement

For finger print enhancement many different methods were tried and tested for good finger print recognition quality. A little loss of information was allowed since with loss of information we could accept similar loss on all images and that made the enhancement procedure uniform. The following were the steps used in enhancement procedure.

Following is the original image that was going to be processed.
The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

This gave us the following image after histogram equalization of the original image.

3) FFT and Adaptive thresholding

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images.

During the threshold process, individual pixels in an image are marked as “object” pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as “background” pixels otherwise. This convention is known as threshold above. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label.

The enhanced image after FFT enhancement has the improvements to connect some falsely broken points on ridges and to remove some spurious connections between ridges and hence is used.

4) Thinning and Cleaning

This process involved cleaning up noise spots and thinning of the image. This was carried out using the matlab commands to morph a black and white image. The output of these operations was as below.
5) Removing small line areas

This technique was used to remove all lines in the figure that were less than a given threshold of points. This allowed removal of noise as well as false minutiae points. This technique reduced the number of minutiae points identified in the later algorithms.

Following are the enhancements of a fingerprint from two different sessions. It can be noted that similar enhancement causes equal loss in data and can be hence treated equally during extraction process.

Final fingerprint images of two different fingerprint images of the same person taken during two different sessions.

We now try to extract the ridge endings and bifurcation points as shown below.

B. Minutiae Points Extraction

After the fingerprint ridge thinning, marking minutiae points is relatively easy. But it is still not a trivial task as most literatures declared because at least one special case evokes my caution during the minutia marking stage.

In general, for each 3x3 window, if the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending.

Here we can see that a 3x3 window can define a bifurcation due to the behaviors of neighboring pixels.

Here we can notice that we have a ridge termination point. The center pixel is marked as a Termination point and the angle at which it is oriented is found with quantization of 45 degrees.

IV. MATCHING ALGORITHMS

Various fingerprint matching algorithms were tested to get a good match between the finger prints. Below we provide the technical details of all these algorithms.

A. Cross Correlation Technique

In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. This is also known as a sliding dot product or inner-product. It is commonly used to search a long duration signal for a shorter, known feature. It also has applications in pattern recognition, single particle analysis, electron tomographic averaging, cryptanalysis, and neurophysiology.

\[
(f \ast g)[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m] \ g[n + m].
\]

For example, consider two real valued functions \(f\) and \(g\) that differ only by a shift along the x-axis. One can calculate the cross-correlation to figure out how much \(g\) must be shifted along the x-axis to make it identical to \(f\).
This is a first simple break through for minutiae points matching. We connect all the minutiae points by a single line in the increasing order of their X axis. This is a signal which we smoothen.

We operate similarly on both reference and template hence we get two curves. We cross correlate them and find the point of maximum matching. This is the cross correlation score returned by our function.

An example of such curves is as given below. We can notice that the two curves match closely in their input signal. Their cross correlation also gives a high value. Here the first one is the Template Image, the second is the Reference image and third is the cross correlation plot.

![Cross Correlation Plot](image)

Here we notice that the number of points in both the input curves is 100. This gives a good results if we set a threshold.

This method fails as we match two templates of unequal length. This is because the cross correlation cause the curves to match at multiple locations and give a high value of cross correlation.

![Cross Correlation Plot](image)

**B. Hungarian Matching Algorithm**

The **Hungarian method** is a combinatorial optimization algorithm which solves the assignment problem in polynomial time and which anticipated later primal-dual methods. It was developed and published by Harold Kuhn in 1955, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Dénes König and Jenő Egerváry.

We are given a nonnegative $n \times n$ matrix, where the element in the $i$-th row and $j$-th column represents the cost of assigning the $j$-th job to the $i$-th worker. We have to find an assignment of the jobs to the workers that have minimum cost.

The algorithm is easier to describe if we formulate the problem using a bipartite graph. We have a complete bipartite graph $G=(S, T; E)$ with $n$ worker vertices ($S$) and $n$ job vertices ($T$), and each edge has a nonnegative cost $c(i,j)$. We want to find a perfect matching with minimum cost.

Let us call a function $y : (S \cup T) \to \mathbb{Q}$ a potential if $y(i) + y(j) \leq c(i,j)$ for each $i \in S$, $j \in T$.

The value of potential $y$ is $\sum_{v \in S \cup T} y(v)$.

It can be seen that the cost of each perfect matching is at least the value of each potential. The Hungarian method finds a perfect matching and a potential with equal cost/value which proves the optimality of both. In fact it finds a perfect matching of **tight edges**: an edge $ij$ is called tight for a potential $y$ if $y(i) + y(j) = c(i,j)$. Let us denote the sub graph of tight edges by $G_y$. The cost of a perfect matching in $G_y$ (if there is one) equals the value of $y$.

We can notice that such an assignment algorithm can be implemented for assigning a minutiae point from the template image to reference image. This is required to be optimal in terms of the distance. However, we might have multiple matches as shown in figure below.

![Multiple Assignments](image)

The above picture shows multiple assignments which can be removed by choosing the best amongst the assigned minutiae points.

Even with such low alignment we get a very good equal error rate as given below.
The dataset used was of 211 images.

C. Edit Distance

In information theory and computer science, the edit distance between two strings of characters is the number of operations required to transform one of them into the other. There are several different ways to define an edit distance, and there are algorithms to calculate its value under various definitions.

We use the Levenshtein distance here for our measure of matching between the reference and the template images. It takes into account the distances of the minutiae points and the angle in which they are aligned.

This method did not give satisfactory results.

D. Neighbor Vector

This method used the neighbor vector and kept the information of the neighboring minutiae points. It strongly matched the neighbors but loosely matched the minutiae points in distance. It also took into account the rotation by using polar angles. This method gave good results and was used in further experimentation.

The Equal error rate plot of this technique is given above.

Further from this we are using either the Hungarian match technique or the neighbor vector method to match our fingerprints.

V. Fuzzy Vault Technique

While fusion of multiple biometric sources significantly improves the recognition accuracy, it requires storage of multiple templates for the same user corresponding to the individual biometric sources. Template security is an important issue in biometric systems because unlike passwords, stolen biometric templates cannot be revoked. Hence, we implement the scheme for securing multibiometric templates as a single entity using the fuzzy vault framework.

The motivation for the fuzzy vault is as below.

A player Alice may place a secret value $K$ in a fuzzy vault and "lock" it using a set $A$ of elements from some public universe $U$. If Bob tries to "unlock" the vault using a set $B$ of similar length, he obtains $K$ only if $B$ is close to $A$, i.e., only if $A$ and $B$ overlap substantially. In contrast to previous constructions if this avor, ours possesses the useful feature of order invariance, meaning that the ordering of $A$ and $B$ is immaterial to the functioning of the vault. As we show, our scheme enjoys provable security against a computationally unbounded attacker.

A. Mathematics behind Fuzzy Vault

Let us briefly sketch the intuition behind our scheme. Suppose that Alice aims to lock a secret $K$ under set $A$. She selects a polynomial $p$ in a single variable $x$ such that $p$ encodes $K$ in some way (e.g., has an embedding of $K$ in its coefficients). Treating the elements of $A$ as distinct $x$-coordinate values, she computes evaluations of $p$ on the elements of $A$.

We may think of Alice as projecting the elements of $A$ onto points lying on the polynomial $p$. Alice then creates a number of random chaff points that do not lie on $p$, i.e., points that constitute random noise. The entire collection of points, both those that lie on $p$ and the chaff points, together constitute a commitment of $p$ (that is, $K$). Call this collection of points $R$.

The set $A$ may be viewed as identifying those points in $R$ that lie on $p$, and thus specifying $p$ (and $K$). As random noise, the chaff points have the effect of concealing $p$ from an attacker. They provide the security of the scheme. Suppose now that Bob wishes to unlock $K$ by means of a set $B$. If $B$ overlaps substantially with $A$, then $B$ identifies many points in $R$ that lie on $p$, so that Bob is able to recover a set of points that is largely correct, but perhaps contains a small amount of noise.

Using error correction, he is able to reconstruct $p$ exactly, and thereby $K$. If $B$ does not overlap substantially with $A$, then it is infeasible for Bob to learn $K$, because of the presence of many chaff points. (If $B$ overlaps "somewhat", then he may still be able to recover $K$. The gap between feasible recovery and infeasible is fairly small, however, as we discuss below.)

The hardness of this scheme is based on the polynomial reconstruction problem, a special case of the Reed-Solomon list decoding problem. This is discussed below followed by the actual implementation in later sections.
B. Reed Solomon Codes and polynomial

Reed–Solomon error correction is an error-correcting code that works by oversampling a polynomial constructed from the data. The polynomial is evaluated at several points, and these values are sent or recorded. Sampling the polynomial more often than is necessary makes the polynomial over-determined. As long as it receives "many" of the points correctly, the receiver can recover the original polynomial even in the presence of a "few" bad points.

VI. FUZZY VAULT IMPLEMENTATION

The implementation required the minutiae points encoded using a polynomial. following reference image can be used as the template for the design principle.

The algorithm is as below.

1. We extract a vector of minutiae points.
2. We take a secret K, and generate a polynomial P with K as its coefficient.
3. The minutiae points extracted are evaluated over a polynomial to get the encoded version of minutiae points.
4. This encoded version is kept along with the minutiae points X coordinates.
5. Fuzzy points are added to the vault to avoid reconstruction of the polynomial.
6. The reference finger print image is processed similarly.
7. The points matching in the template and reference finger print are plotted.
8. These points are interpolated to get a curve.
9. The coefficients of this curve are the secret key K.

The following figure shows the fuzzy chaff points as generated while testing. It minutely shows extra dots on the figure where the finger print lines do not exist. These points are extra random points generated in that square.

In case the reference template does not substantially overlap with the template curve, the polynomial is not reconstructed.

In case of more number of matched points we need to take all combinations of size K from the matched points.

The coefficients of the polynomial might not exactly reproduce the secret key K. This is because the finger prints do not exactly overlap. This problem is solved by using reed Solomon codes to error correct the polynomial. Incase its not possible to recover the key and authentication fails.

Following is the diagram of this security infrastructure.

The implementation was tested over a subset of images manually and gave the following ERR curve when interpolated.

It gave very low FRR at the ERR yet its highly unsatisfactory and not comparable to the results given by the authors of the original paper.
VII. PROPOSAL OF EXTENDED SECURITY USING FUZZY FRAMEWORK

The above system can be used to combine the knowledge based security approach with biometric decoding of the key. This ensures unique key for each biometric template and might guarantee 100% GAR with 0.00% FAR. The proposal follows the following flowchart.

![Flowchart Image]

It can be noted that the hash of the key is transmitted to the system so that the original biometric key knowledge will not be decipherable and reproducible.

VIII. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

ACKNOWLEDGMENT

I thank Prof. Phalguni Gupta for constant guidance and motivation to take up a project in this domain. He has given us a great opportunity to move into the field of biometric identification. I thank my mentors Surya Prakash, Badrinath, and Umarani for their support during the course of project. Any difficulties were explained at length and really helped in completion of the project. I thank Anubhav Rajesh and many other batchmates who gave constant encouragement and helped with creative ideas.

REFERENCES

[2] A Fuzzy Vault Scheme, Ari Juels and Madhu Sudan
[3] Fingerprint-based Fuzzy Vault: Implementation and Performance Karthik Nandakumar, Student Member, IEEE, Anil K. Jain, Fellow, IEEE and Sharath Pankanti, Senior Member, IEEE
[6] Reed-Solomon Codes by Bernard Sklar
[7] Local Features for Enhancement and Minutiae Extraction in Fingerprints Hartwig Fromhals, Klaus Kollreider, and Josef Bigun, Senior Member, IEEE
[9] A Novel Algorithm for Detecting Singular Points from Fingerprint Images, Jie Zhou, Senior Member, IEEE, Fanglin Chen, and Jimwei Gu, Student Member, IEEE