FORESTFINGERS: MULTI-FINGERPRINT RECOGNITION WITHOUT SEGMENTATION

by

ABHIJIT ZIPRU BENDALE

B.E University of Pune, 2007

A Thesis submitted to the Graduate Faculty of the

University of Colorado at Colorado Springs

in partial fulfillment of the

requirement for the degree of

Master of Science

Department of Computer Science

2009

This thesis for Master of Science degree by

ABHIJIT ZIPRU BENDALE

has been approved for the

Department of Computer Science

by

Terrance Boult, Chair

Prof. Rory Lewis

Prof.Xiaobo Zhou

Bendale, Abhijit Zipru (M.S., Computer Science)

ForestFingers: Multi-fingerprint Recognition without Segmentation

Dissertation directed by El Pomar Professor Terrance Boult

Multi-fingerprint matching is an important problem for the biometrics community. To increase collection speed and decrease potential false matches, many identity management programs have moved to capturing Slap images, were a single image simultaneously collects images of four or eight fingerprints. Multi-finger matching is traditionally done by segmenting the slap images, to isolate fingertips and applying individual fingerprint matching to the segmented results. Though this is approach is coherent with the notion of backward compatibility, the process of segmentation has numerous problems. We propose a novel method of multi-fingerprint recognition without segmentation. In our approach we create "forests of trees" from minutiae pairs in the fingerprint, forming consistent connected components in the forests. The size of these consistent connected components determines the match score. Since this representation does not require any segmentation of the fingerprint slap data into individual fingers, it is more robust to spatial, rotational and other variations and can make use of added data from other segments of the finger. The approach presented here is a true mul- tiple fingerprint matching approach as opposed to fusing matching re- sults from individual fingers. The Forest Finger algorithm can be applied to multiple independent fingers without finger assignment. Our results on the NIST DB29 shows superior performance when compared with the existing NIST Bozorth matcher [3] applied individually to segmented prints. In the latter section of the thesis, we also discuss ways of extending this algorithm to to mix data from multiple-fingers, making it infeasible to search the database with latent prints from single finger. We argue that this approach is necessary for the purpose of building application specific databases, a notion important for protecting privacy of an individual. This thesis provides an approach, that can be followed by many existing methods to extend their methods for performing "true" multi-fingerprint matching.

Acknowledgements

I consider myself extremely fortunate to have had an opportunity to work with someone like Prof. Terrance Boult. Terry has been a great mentor and a constant source of motivation for over two and a half years. I am thankful to Terry for first granting me the freedom to find my own way of working and later for pushing me hard to extract the best out of me. Terry has been a remarkable influence both on personal and professional front and I will always be indebted to him. I cannot thank Terry enough for keeping me funded during entire duration of my masters. Terry went out of his way to support my internship at MIT and I am extremely grateful to him for that. I hope this is just the beginning of our collabration in research.

Anthony Magee, who has been my friend, labmate and roomate deserves special mention. He helped me in the early days to get me upto speed in programming and computer science in general. He made my transition from electrical engineering to computer science as painless as possible. Most of all, he has been a great friend and has kept my spirits high during the times I felt down. Walter Scheirer has been a great senior student in the lab. He has always had encouraging words for me and has been a great collabrator.

On social front, I am extremely thankful to Ginger Boult for her kindness and generosity. I will always cherish the trips to Copper Mountain, Pikes Peak, seven falls and

Canon City. She has been very encouraging from the start and has lit up many a conversations with her brilliant sense of humour. Special thanks to Anderson, Ana and Ashish Gupta!! Friends in Colorado like Archana, Palden, Kunal and members of VAST lab have been extremely helpful to me in numerous ways and I think they rightly deserve a mention here.

Nicolas Pinto, my supervisor at MIT has been instrumental in shaping my thinking. I am thankful to him to have allowed me to work remotely at the same time being very considerate about my schedule, during my internship in MIT. I would also like to thank Prof. James DiCarlo for hosting me for my internship at MIT. I am thankful to Prof. D G Khairnar for teaching me how to cradle big dreams while keeping one's feet firmly on the ground.

Most importantly I would like to thank my mom, dad and sister who have always encouraged me to do my best. My dad has been a constant source of inspiration and has taught me the true meaning of perfection. I shall always aspire to be as good as him in technology. My mom, for always being there for me, for listening to me, and for protecting me from my sister and my dad. Most importantly I thank her for all the sacrifices she has made along the way. My sister has been a great listener, motivator and supporter. She is the most courageous person I know.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Biometrics

Identity management lies at the very core of efficient functioning of any country. Biometrics have gained prominence in recent years as a premier identity management tool. Biometrics helps to identify a person based on his biological(anatomical or physiological) and/or behavioral characteristics. Various biological biometrics such as fingerprints, face, voice, iris have gained importance in recent years. A good biometric is universal, unique, does not change significantly over time and is easy to collect. Fingerprints, since the seminal work of Francis Galton, has been widely used in forensic investigations because of their strong uniqueness. However, in last decade biometrics and fingerprints specifically, have been found to be useful not just in forensic investigations but also in many identity management programs like National ID programs, passport programs, and more recently even commercial domains like access control at Disneyland.

As fingerprint recognition systems become more and more widespread, its performance needs to match the scale of population size that it intends to cater. Many automated fingerprint recognition systems like IAFIS [2] have been making significant progress in past decade or so to improve the identification rates and reduce false match rates in large scale fingerprint identification and verification systems. However, even though the systems are getting more and more accurate there are computational and algorithmic limitations. A recent study conducted by Executive office of the President [1] concluded that as the number of fingerprints captured per individual increases, the accuracy with which he can identified significantly increases. In resonance with this theme, multi-fingerprint recognition has been of interest to US Citizenship and Immigration Service (USCIS), Federal Bureau of Investigation (FBI) and National Institute of Standards and Technology (NIST), Department of Justice (DOJ) and Department of Homeland Security(DHS) as well as many international organizations.

1.2 Multi-Fingerprint Recognition

Slap fingerprints, or simultaneous plain impressions, are a single image that capture the fingerprints of multiple fingers at the same time. These can be captured by a live scanner

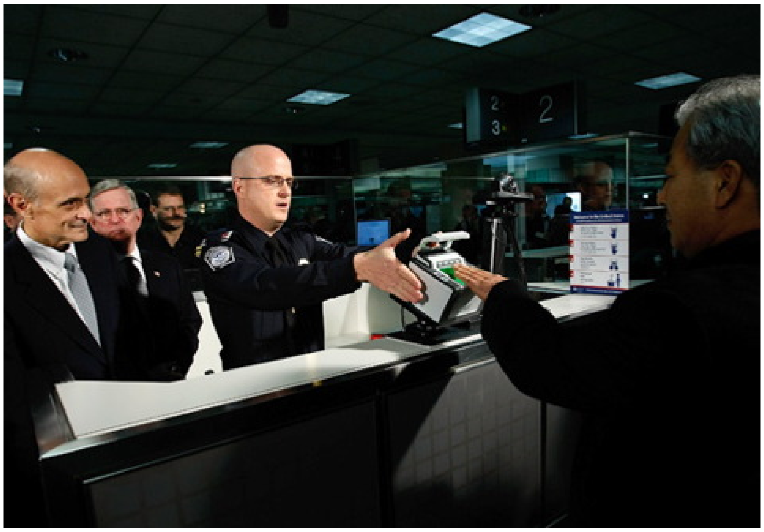


Figure 1.1: Example slap scan at a US-VISIT counter at an airport. Slaps are quick and easy to capture and at the same time capture significant amount of information about the individual (Picture taken from [19])

or can be ink prints [16]. Figure 1.1 shows an example capture of multi-fingerprint capture at a DHS counter.Slap fingerprints are compromise between rolled finger-prints and single-finger flat fingerprints. Slap fingerprints are more tolerant to attempts of spoofing they system(by changing the order of enrollment of the finger) since all the fingers have to be scanned at the same time. Capturing of slap images is easier, faster and much less error prone than rolls or single finger flats.

1.3 To Seg or Not to Seg

Fingerprint recognition has, since its earliest days, presumed segmentation of any multi-finger data. Part of that was pragmatic: to reduce computational processing. Part of it was effectiveness, when data was hand segmented and hand labeled. Early techniques were sufficient on scanned fingerprint cards and since early digital fingerprint sensors were/are single finger sensors, so segmentation was not a significant issue. Post 2001, to address the speed of processing, and reduce potential of ordering errors in large scale programs, such as boarder crossing or other large-scale biometric identification programs, a wide range of "slap" sensors have been developed that capture multiple fingers or even whole hands at once, e.g. see figure 1.4. To maintain backwards processing compatibility, the community worked to develop automated segmentation algorithms to extract the distal phalanges (fingertip) images from the slap, with some example successful and unsuccessful segmentation (from NIST software) overlaid on the slap image. While it may be expedient to segment first to reuse old fingerprint algorithms, prepossessing with segmentation means that any error in the segmentation will negatively impact the recognition, and in this case it also means throwing away data from the intermediate phalanges. Slap Segmentation was evaluated by NIST in [17], and while good, was not perfect, concluding " The most accurate segmenters produced at least three highly matchable fingers and correctly identified finger positions in from 93% to over 99% of the slap images,

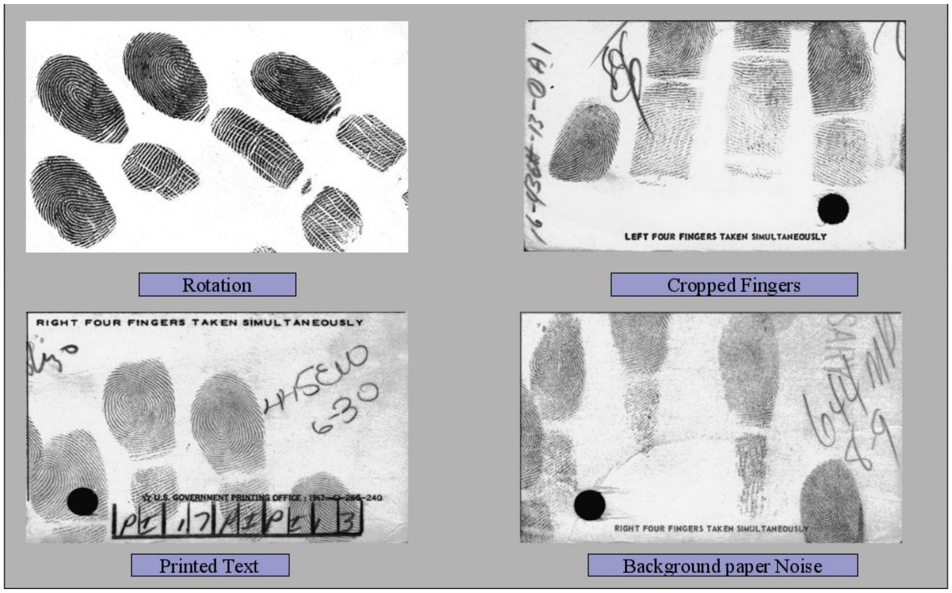

Rotation
LEFT FOUR FINGERS TAKEN SIMULTANEOUSLY
Cropped Fingers
RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY
U.S. GOVERNMENT PRINTING OFFICE
Printed Text
RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY
Background paper Noise

Figure 1.2: Segmentation process is very senstitive to variations commonly found in slap images

depending on the data source.” When working on recognition problem with potentially 1Billion (100Million*10) or more fingers in the database to match against, a percentage point in accuracy is very significant.

Segmentation algorithms were only moderately successful in segmenting slaps into individual fingers because of inherent variations in slap images. Variations in orientation, significant amount of paper noise or background noise, existance of printed text, cropped fingers caused significant problems. Figure 1.2 illustrates this point in further detail.

Not only that segmentation process was sensitive to variations, many times it captured wrong piece of data and identified it as a finger. Refer the example in figure 1.3

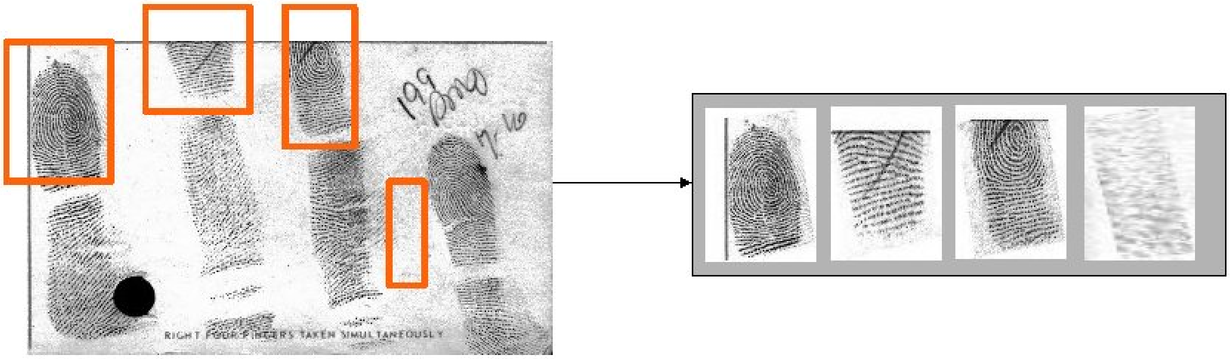


Figure 1.3: Many times segmentation failed miserably capturing completely random information as fingerprint

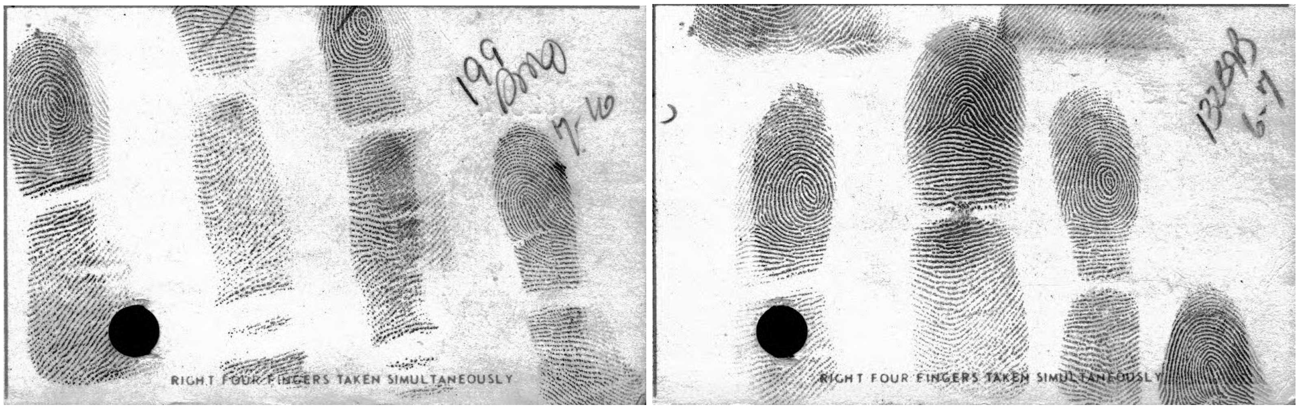

RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY
RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY

Figure 1.4: Example matching Slap Images from NIST Special Database 29 [16], with the sub-regions detected by NIST slapseg overlaid in red.

1.4 Contribution of the work

Hence we question the notion: Is it necessary to segment the image in order to recognize/match the fingerprint or can we completely bypass the segmentation process?

The contribution of the work is 2 fold. First, we introduce a method of matching multiple fingerprint without segmentation. We call this algorithm as ForestFingers algorithm. In our approach we create forests of trees from edges of minutiae pairs, forming consistent connected components in the forests. The size of these consistent connected components determines the match score. Since this representation does not require any segmenta- tion of the fingerprint slap data into individual fingers, it is more robust to spatial, rotational and other variations and can make use of added data from other segments of the finger. Our results on the NIST DB29 shows superior accuracy when compared with the existing NIST Bozorth matcher applied individually to segmented prints, to fused rolled prints, or applying Bozorth directly to the slap images.

The second contribution of this work is related to privacy. We introduce the concept of id-privacy and show how using forest-representation using unsegmented data what may be the single most important "privacy" issue in biometrics: how to prevent function creep in large-scale biometric programs. We show we can achieve 2-id-privacy for fingerprint-based recognition allowing de-duplications while preventing searching with a latent print.

Chapter 2

ForestFingers: Multi-Fingerprint Matching without segmentation

2.1 Prior Art

A fingerprint matching algorithm compares two given fingerprints and returns either a degree of similarity or a binary decision (mated/non-mated). Without the loss of generality, we denote the input fingerprint to be matched as probe against a gallery (database) images. We are more interested in the fingerprint verification problem (i.e. searching for an input fingerprint in gallery of N fin- gerprints). Most of the current fingerprint matching approaches can be divided into correlation-based matching (Stoianov et al (1999), Watson et al (2000)), minutiae based matching and non-minutiae fea- ture based matching systems. There has been significant progress in minutiae based fingerprint matching systems. Minutiae based matching methods are of particular interest because these methods have often been the best performing methods on many of the current benchmark tests like FVC 2002, FVC 2004 and FVC 2006. Among these methods, people have focussed on geometric methods, Hough Transform based methods, minutiae matching with pre-alignment. Also in these methods there have been some more methods focussing on local minutiae matching where emphasis is on matching local minutiae structures which are invariant with respect to global transformation (e.g. translation, rotation etc)and hence suitable for matching without any a priori global alignment.

For the problem of simultaneous multiple-fingerprint matching without segmentation, the afore mentioned methods have some drawbacks. Correlation based methods have been abandoned by the fingerprint recognition community because of the costly hardware involved and its lack of performance. These methods also suffer significantly from rotation and distortion variations. Minutiae based local structure matching methods have gained prominence in recent past because of its performance. These methods capture local structure like stars Ratha et al 2000, triangles Kovacs et al 2000, Bhanu et al or exhaustive graph search like Kplets (Govindraju et al) or Bozorth Matcher (2003)). This local structure matching stage is followed by a consolidation stage, where usually maximum size local structure is quantified by a match-score as a measure of similarity between the fingerprints. These methods have shown good performance on single fingerprint recognition problem, however they lack the notion of **distributed** consolidation of the local structure in different parts of image which is essential for the recognition of multiple fingerprints. What is needed here is a representation that is invariant to translation and orientation and allows formation of distributed structures in different parts of the image (ideally a separate representation localized around each fingerprint in the image). If we have such a representation, it would be independent of number of fingers in the slap image hence it can be easily adapted as per need to any number of fingerprints in the image.

2.2 Bozorth Matcher: Quick Overview

The algorithm that we propose in the following section is loosely based on, and compared to, NIST Bozorth Matcher, which is a minutiae based local structure matching method for single fingerprint matching [**?**]. Minutiae points are interest points in fingerprint image which indicate points of ridge bifurcation or ridge ending, sometimes also termed as level 2 features of fingerprints. Use of minutiae points for fingerprint identification has long history in forensic investigation , which is also one of the main reasons for use of these features in automated fingerprint identification systems.

We provide here a brief discussion of Bozorth matcher. The reader is advised to check the references for more details. The Bozorth matcher works on minutiae files created by MINDTCT program of the NFIS2 package [**?**] which is a list of minutiae points in a fingerprint image with (x, y, θ, q) entries where (x, y) is the position of minutia point in

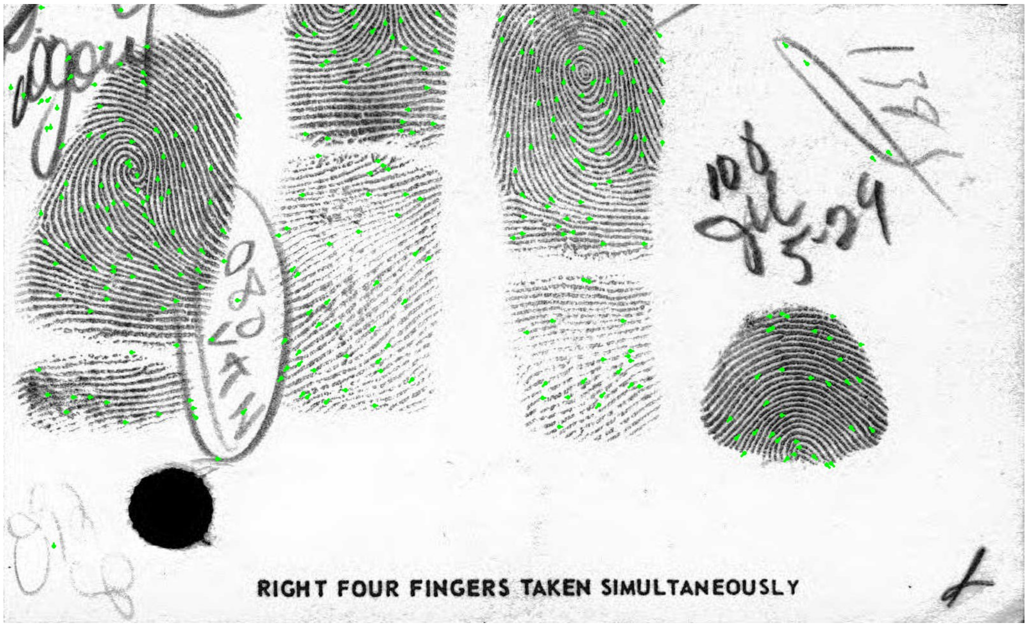


Figure 2.1: Minutiae points detected and highlighted on a fingerprint image

fingerprint image, θ is the orientation angle of the ridge at minutia point and q is the quality of the minutia point determined by NIST NFIQ algorithm. Since this is a fingerprint matching algorithm it considers probe and gallery image minutiae files simultaneously. Refer figure 2.1 to understand what minutia file represents.

The bozorth algorithm is as follows [3]:

1. **Construct Intra-Fingerprint Minutia Comparison Tables:** One table for the probe fingerprint and one table for each gallery fingerprint to be matched against

2. **Construct an Inter-Fingerprint Compatibility Table:** Compare a probe prints minutia comparison table to a gallery prints minutia comparison table and construct

a new compatibility table

3. **Traverse the Inter-Fingerprint Compatibility Table:** Traverse and link table entries into clusters and Combine compatible clusters and accumulate a match score

The major drawback with Bozorth matcher in terms of handling multiple fingerprints is that it forms a single maximally connected and consistent graph/web for entire fingerprint. Hence, when considered for a slap, it would form a representation for a single fingerprint only. Here we bring in key novelty into Bozorth matcher with our approach. Figure 2.2 shows the entire pipeline of our approach with pointers as to where we start building on Bozorth matcher. The following sections describe in detail the entire process and in every section we will highlight our contributions.

2.3 Pair Table Formation

The first stage in the algorithm is Pair-table formation. Let us denote list of minutiae points from probe as $P = \{m_1, m_2, ..., m_p\}$ where $m_i = \{x_i, y_i, \theta_i\}, i = 1...p$ and minutiae points from gallery as $G = \{m_1, m_2, ..., m_g\}$ where $m_j = \{x_j, y_j, \theta_j\}, j = 1...g$. From sets P and G relative measurements (distances and angles) are computed from each minutia to all other minutiae points of the respective fingerprint (termed as intra-fingerprint minutia comparison tables in NFIS2). P will yield pair-table $PT_p = \{p_1, p_2, ...p_q\}$ where $p_i, i = 1..q$ is an entry in PT_p, for each probe image. Similarly, G

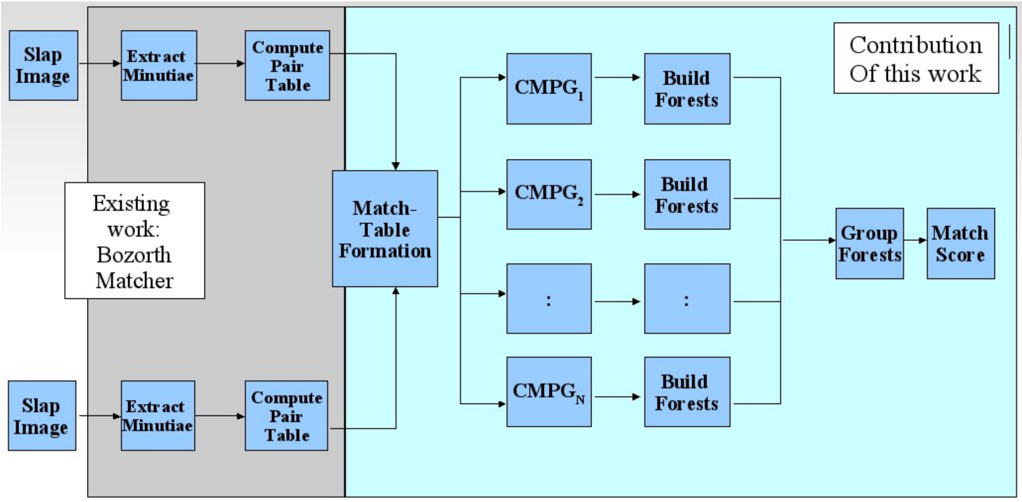


Figure 2.2: ForestFingers Algorithm Pipepline. We build on existing Bozorth matcher and add our own set of modifications that enable us to do multi-fingerprint matching without segmentation

will yield pair table $PT_g = \{p'_1, p'_2, ...p'_r\}$ where $p'_j, j = 1..r$ is an entry in PT_g for each gallery image. Every entry in PT_p and PT_g consists of $\{d_{kj}, \beta_1, \beta_2, k, j, \theta_{kj}\}$ where $d_{kj}$is the relative distance, $\beta_1, \beta_2, \theta_{kj}$ are relative angles and k, j are indices of the minutiae points under consideration. The pair tables are prunned by a (relative) distance threshold (the reason being to have emphasis on local structure analysis). To understand what each edge means with respect to fingerprint refer figure 2.3

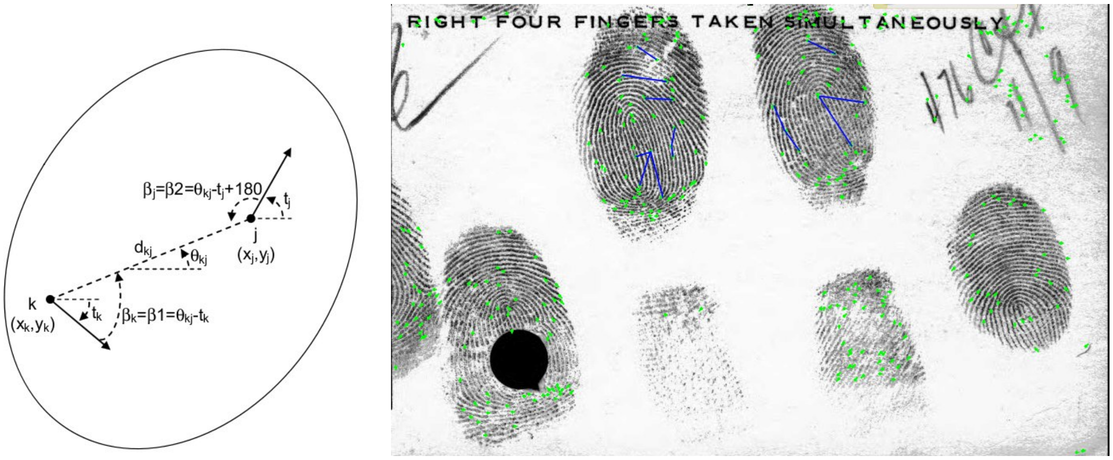


Figure 2.3: Pair-table consists of computation of relative distances and angles between minutiae points making the representation rotation and translation invariant. In the above figure shown is the concept of computation of relative distances and angles between minutiae points (figure from from [3]) along with its physical interpretation on actual fingerprint (the edges are displayed in blue) This figure best if viewed in color

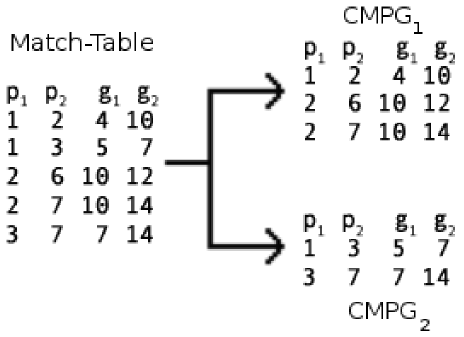
2.4 Match-Table Formation

In next step, PT_p and PT_g are sorted on distance, and each entry of PT_p is compared with each entry of PT_g to generate a list of compatible entries where two entries of PT_p (p_i) and PT_g (p'_j) are compatible if and only if the distances and angles are within prespecified tolerances. This list of potentially compatible entries between PT_p and PT_g form match table $MT = \{r_1, r_2, ...r_m\}$ where $r_i, i = 1...m$ is each compatible entry in the match table and m is the total number of entries in the match table. Each entry i.e. r_i in MT consists of $\{\Delta_\beta(\theta(P_m), \theta(G_n)), k(P_m), j(P_m), k(G_n), j(G_n)\}$ where one pair is from probe fingerprint $(k(P_m), j(P_m))$ and other is from gallery fingerprint $(k(G_n), j(G_n))$. Thus, we have four minutia points per entry in match table (two from probe and two from gallery. Match-table is termed as inter-fingerprint compatibility table in NFIS2. Each entry in match-table thus has correspondence information: which means k^{th} entry of probe $k(P_m)$ potentially matches to k^{th} entry of gallery $k(G_n)$ and similarly for the j^{th} entry in probe $j(P_m))$ corresponds to j^{th} entry in probe $j(G_m))$. This is based on the hypothesis that for a pair from probe and gallery, if the relative distances and angles are within tolerance, then those pairs are potentially the same pairs (i.e. they represent the same minutiae points in probe and gallery image)

We introduce our own set of modifications to the basic Bozorth process. During the formation of pair-table, we also include a field with the the product of minutiae qualities under consideration to the pair-table. While forming the match-table, if the edge-pairs are compatible, we add a “quality score” in the match-table, which can be raw minutiae quality or can also include differences between fields. The intuition for this modification is that we want to favor the minutia points/pairs with higher quality. In the discussion section, we argue that this quality reward not only helps to increase the accuracy of the algorithm, but also gives significant advantage to the algorithm in terms of time of execution.

2.5 Formation of CMPGs

The correspondence information in match-table is important since the match-table created in previous section contains many inconsistent assignments. Consider figure 2.4, which contains an example of match-table entries. In this figure the column p_1 corresponds to column g_1 (i.e. minutia point in p_1 (probe) is same as minutia point g_1 (gallery)) and similarly p_2 corresponds to g_2. The first entry (row of an example match-table) says minutia point 1 in probe matches with minutiae point 4 in gallery. However, the second entry in the match-table contradicts it saying minutia point 1 in probe corresponds to minutia point 5 in gallery. We obviously need to remove such inconsistencies. For this purpose we separate the match-table into set of consistent minutia point assignment groups which we call Consistent Minutiae Pair Groups or CMPGs. The process of separating the match-table into CMPGs is shown in figure 2.4. Figure 2.4 shows formation of two such CMPGs. Here note that each minutia point in the probe columns of the CMPG, has one and only one corresponding minutia point assigned in gallery columns of the CMPGs. This is a greedy process because we start from the first entry in the match-table and find all the successive entries in the match-table that are consistent with the current CMPG (i.e. there is one and only one probe minutia point to gallery minutia point assignment). As soon as an inconsistency is encountered (as in row 1 and row 2 of match-table) a new CMPG is created. This process is continued until all the rows of match-table are members of their

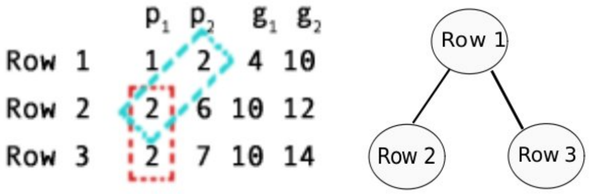


p_1	p_2	g_1	g_2
1	2	4	10
1	3	5	7
2	6	10	12
2	7	10	14
3	7	7	14

p_1	p_2	g_1	g_2
1	2	4	10
2	6	10	12
2	7	10	14

p_1	p_2	g_1	g_2
1	3	5	7
3	7	7	14

Figure 2.4: Inconsistencies in Match-Table and formation of Consistent Minutiae Pair Groups (CMPG)



	p_1	p_2	g_1	g_2
Row 1	1	2	4	10
Row 2	2	6	10	12
Row 3	2	7	10	14

Figure 2.5: Formation of links in a CMPG. Here we say that row 1 and row 2 of match-table are form a link between one another, since they have a minutia point (2 for probe or 10 for gallery) in common

respective CMPGs. Empirically we have found that majority of rows of match-table get grouped within first 20 or so CMPGs, for true-match (i.e. probe and gallery images are from the same person).

2.6 Formation of Links

Referring to Figure 2.5, we say that a particular row in match-table is connected to another row if and only if there exists a common minutia point between them. Consider $CMPG_1$ in our example from 2.4 In this example, 2 is the common minutia point between row 1 and row 2 (and also row 2 and row 3) in probe part of the match-table. Similarly, 10 is common minutia point joining row 1 and 2 in gallery part (and also row 2 and row 3). In this way we proceed to find all the connections a row can have in a CMPG. We term these connections as links within rows of CMPGs. You may have noticed in the CMPG that each pair in probe part of match-table is consistent i.e. a pair in probe will have its corresponding consistent pair in the gallery. Formation of links in probe part of the match-table is equivalent to formation of links in gallery part of the match-table. This is a brute-force method since in a CMPG all the possible links are searched and grouped.

2.7 ForestFingers

In the previous subsection, we saw how links are formed within different rows in CMPGs. It is important to understand what these connections mean in terms of fingerprints. When we find connection between two rows of CMPGs, we actually find a path that connects minutiae points on the fingerprint. Consider Figure 2.6. In this figure, minutia point 2 was common between row 1 and row 2. This means that two edge-pairs formed from minutiae points (1-2) and (2-6) had a connecting path between them. Same is true in case of row 2 and row 3 of the example CMPG shown in 2.6. Thus, we have a small cluster of connected minutiae points on fingerprints that have their relative distances and angles matching (within certain tolerances). As the size of such connected components goes on increasing, it would mean that more and more minutiae points have the distances and angles between them matching exactly. This would be possible only if they were actually same parts of fingerprint, which means that by traversing a path through entries of CMPGs we are capturing local structure of the fingerprint, and as the size of the path traversed increases, we are proceeding towards capturing significant amount of common structure between probe and gallery fingerprints. It is also important to note that it is possible that in different parts of CMPGs or across different CMPGs we might not find a single connection that is big enough to capture entire fingerprint or all the fingerprints (in case of slaps). However, as long as these connected component form clusters big enough to capture "significant" amount of local structure, we should allow formation of such clusters in different parts of fingerprints.

Here each row in the match-table can be viewed as vertex of a non-directed graph and the links formed between the rows (as shown in 2.5) as edges of this non-directed graph. Thus, the problem at hand reduces to finding maximum size sub-graph that is consistent in terms of minutia assignment between probe and gallery. Finding connected components

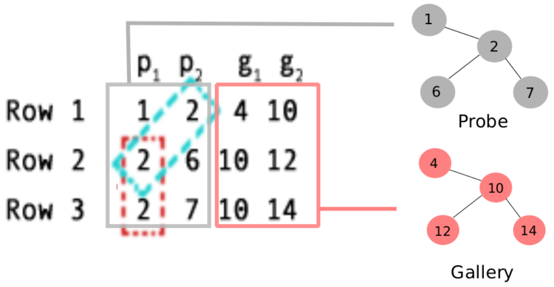


Figure 2.6: Formation of forests in a CMPG

in a graph has rich and long history in computer science, and it has been noted that many methods for graph traversal like depth first search, breadth first search, mazes etc can be used. However, we are not just interested in finding connected components, but at the same time, merging two reasonably sized connected components to capture multiple local structures across fingerprints. These operations an be performed efficiently with union-find algorithm [18]. We term clusters of connected components as trees, and the set of all such clusters as forests of trees.

2.8 Computation of Match Score

In biometrics, match-score quantifies the similarity between input and the database template representations. In previous section we saw how clusters of consistent connected components i.e. forests of trees are formed. We also noticed, how these clusters or trees represent local structure of the fingerprint. However, there is also a very high possibility of matching random edges (i.e. small clusters were found to be matching) which might be more of a chance than capturing actual similarity between fingerprints. Hence, we introduce the notion of considering only those clusters that have at least a certain number of edge-pairs connected to each other. Thus, in 2.6, the size of the tree is 3 (since 3 edges are connected to each other). Higher number of such consistent connected components, more local structure we would have captured in various parts of fingerprint (or across multiple fingerprints in case of slaps). In order to compute the overall similarity between probe and gallery image under consideration, we need a way to quantify a measure of similarity from the matching local structure. For simplicity, we consider the total number of edge-pairs across all CMPGs that are consistent with each other as the measure of similarity between the two fingerprints. Hence, a match score M between probe p_i and gallery g_j means that there were M number of matching-edges in probe and gallery fingerprint that were completely consistent with one another (i.e. a minutia point in probe corresponded to one and only one minutia point in gallery) and formed clusters of connected components.

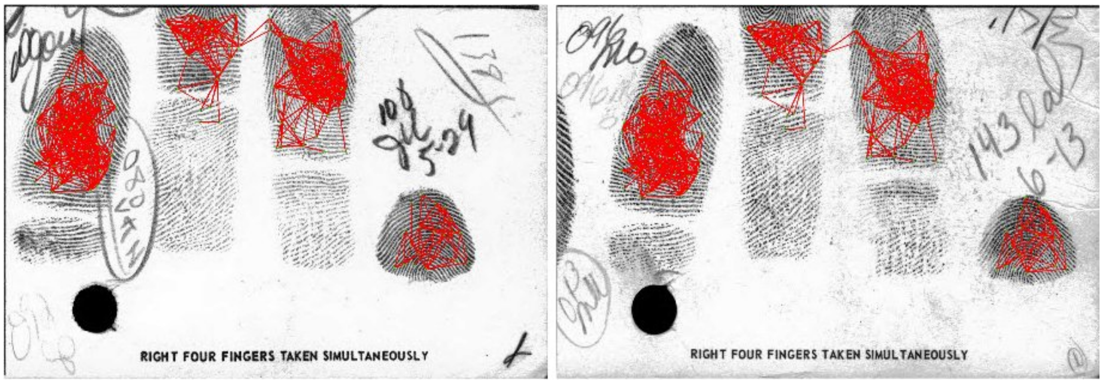

RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY
RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY

Figure 2.7: The above image is an instance of ForestFingers applied on slap images from NIST DB29. (to the left is probe image and to the right is gallery image for a true match). Match-score is total number of edge-pair entries that form connected components (forests of trees). The green points are the minutiae points (best if viewed in color), red lines are pairs that were matched.

Chapter 3

Evaluation of the Matching algorithm

3.1 About the Playground

We use NIST29 database [16] for our experiments. The database is made of paired fingerprint cards that include all ten rolled fingerprints and the plain/flat impressions. There are two such sets of fingerprint cards (a*.an2 and b*.an2) for one individual captured at different dates. The database has 216 paired fingerprint cards each scanned at 19.7 ppmm (500 ppi). The images are compressed using WSQ compression at a compression ratio of 15:1 and stored in the ANSI/NIST data format. The card consists of impressions of individual fingers and also a four-finger impression (slap) of left and right hand per card. There are 216 fingerprint cards for probe (a*.an2) and 216 fingerprint card for gallery(b*.an2). Since, each fingerprint card has 2 four-finger impressions (left and right), our database consist of 432 images in probe side and 432 images in gallery side.

Before using our multi-fingerprint matching algorithms, we extract minutiae points from the image using MINDTCT [3] and store them in minutiae files. The minutiae files contain minutiae points in the form of (x, y, θ, q), where x, y are the position θ is the orientation and q is the quality of the minutia point determined by the MINDTCT algorithm.

3.2 Evaluation of ForestFingers

In order to evaluate our method, we performed experiments with ForestFingers on unsegmented slap images and compared them with NIST Bozorth matcher applied on segmented slap images. For getting performance of NIST Bozorth matcher (scores) on slap images, slap images from NIST DB29 were first segmented using NIST's NFSEG package followed by minutiae extraction process using MINDTCT. Each segmented individual fingerprint from probe was compared with respective segmented fingerprint from gallery (i.e. index finger from probe was compared with index finger of gallery to get score. Similar process was carried out for remaining fingers and final score was addition of scores from comparison all the fingers). For ForestFingers, minutiae points were extracted from unsegmented slap image and ForestFingers algorithm was used for probe an gallery and score was computed as discussed previously. The results of all the three experiments mentioned above are summarized in figure 3.1 as receiver operating characteristic (ROC)

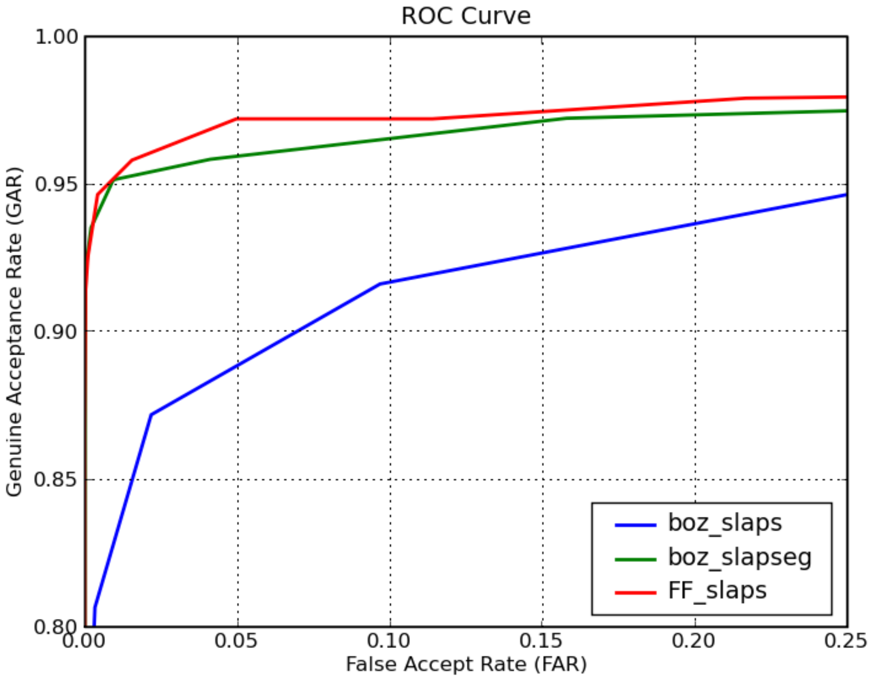


Figure 3.1: ROC curve comparing NIST Bozorth Matcher on segmented images with our ForestFingers algorithm on unsegmented slap images. In spite of not segmenting the data, ForestFingers outperforms Bozorth Matcher. Performance of Bozorth Matcher when applied on straight slaps degrades significantly indicating its dependence on costly segmentation process

curve plotting the genuine accept rate (GAR) against false accept rate at various thresholds.

Chapter 4

New Directions: Cross finger Matching

4.1 Fingerprint, Privacy and Large Scale Identity Management

In the previous sections we discussed our approach of segmentation less multi-fingerprint matching. However, the root cause for the said approach towards recognition was not that of segmentation, but a much larger issue of privacy protection of individual in large scale fingerprint based identity management systems. For large government program, it is critical that they should be able to solve the de-duplication problem, to ensure one ID per person. The big privacy problem is that all existing ways to searching for duplicates also supports searching that database for whatever reason the system owner chooses. Just function creep transformed, the social security number, into an identifier used and abused in ways never imagined when it was introduced.

There is growing privacy concern about biometrics [5] and there is is a rapidly growing body of research developing techniques that convert the raw biometric data into secured non-invertible tokens, with a wide range of techniques now developed including Tuyls et al [6] Boult et al [7], [8], and Nandakumar et al [9, 4].

While these papers present important research, from a privacy point of view it is not sufficient to just protect the template. One of the major fears is function creep, and the potential for the data owner to use the data for other purposes. In particular for government system, which is where the largest systems are being developed/deployed, a serious concern is searching using latent or otherwise obtained fingerprint data to identify the individual. While some argue that the only the guilty have to fear, a non-trivial concern is false identification as in the widely reported case of Brandon Mayfield[10] who was falsely imprisoned based on one latent print. Protected templates don't solve that problem as they still provide for the system owner to "search", which means they can still be used to identify (or misidentify) by searching finger-by-finger through the DB for potential suspects. (Two approach ([7, 4]) provide for password enhanced "verification" only approach, but they cannot be used for de-duplication. De-duplication inherently means "recognition", leading us to ask "is there a way to support de-duplication and yet ensure the recognition data cannot not abused?".

4.2 id Privacy

id Privacy is formally defined in [**?**]. We provide mention the concept here for the sake of completeness. id Privacy in the context of biometrics is defined as [**?**]

Definition 1. *A recognition problem is said to have* id-privacy *when it impossible to use the stored representation to recognize the subject, using only* $i - 1$ *items of input, with probability* d *over random chance, but when when* i *or more distinct inputs are present the subject can be recognized at substantially above chance. For simplicity in the ideal case* $\underset{.}{=}0$*, we refer to this a* id*-privacy, e.g. if it takes 2 independent items to identify the user at all, then we would call it (2,0)-*id*-privacy or simply 2-*id*-privacy.*

When $d = 0$, this is not a statement about an algorithm, but of the problem and the stored data – it must be the case that no algorithm can recognize using the stored data and less than M inputs. This is in spirit quite similar to the secret sharing problem of Shamir [11], but differs in that its not sharing data, but storing data for future recognition and must deal with the approximate nature of recognition problems. This differs from, and is much stronger than, the k-anonymity [12] and related privacy concepts in that we require the recognition is no better than a factor of d above random chance, i.e. if d=0, its full anonymity within the dataset. The focus here is on extremely ambiguities, where it is better to talk about the fraction of the population, not a particular value of k. The other important difference is that, we bound the amount of data the adversary has about the subject to be recognized. Our goal is balancing the ability to preserve privacy while still supporting recognition.

For our problem we want to consider an “input” to be an image of a fingerprint, or part thereof, and we want to ensure that with an image of a single finger (e.g. latent), the person cannot be recognized at all, i.e. we want to ensure 2-id-privacy fingerprint recognition. This paper presents an approach wherein data from slap-images are combined in such a way that they can support de-duplication but such that any single-finger (e.g. latent) cannot be matched at all. We also describe an extension that allows single fingers to be used with a match-rate low-enough to make it impractical for anyone to use it that way.

4.3 Cross Finger Matching

Latent prints are partial prints collected usually from crime scenes and are very helpful cue for forensic investigations. These fingerprints are usually parts of fingerprints left behind by a suspect. However, fingerprint identification algorithms are far from being perfect, and even the best performing automated fingerprint identification algorithms have some false accepts even when they are operating at very low false accept operating rates. In Brandon Mayfield case the important privacy issue was not that his fingerprint already existed in records of FBI from 1984 burglary case in which he was involved, but more importantly that with the help of latents, forensic investigators were able search an

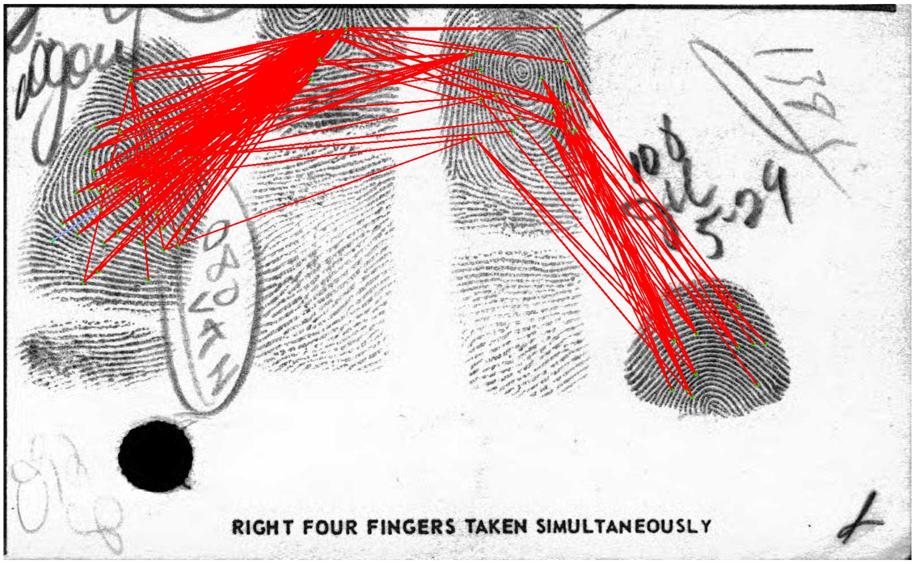


Figure 4.1: The above figure shows formation of forests of trees that span multiple fingers

existing fingerprint database. In the context of national ID programs, the key privacy issue is that if the multi-fingerprint recognition proceeds in segment then match form, then there is no way to restrict "searching" the database based on latent and it is solely at the discretion of the database owner as to how to use it or "abuse" it.

With the above discussion in mind, two primary requirements of a large-scale multi-fingerprint based identity management system: first it should allow the possibility of duplication detection i.e. the system should answer the question with very high accuracy as to is the person already enrolled in the database. It is important for a system like National ID program, so that no individual is assigned more than one passport or similar document. Another, more fundamental question is the issue of privacy, i.e. given a huge National ID multiple fingerprint database, the database should not be used for any other purpose than it is meant to serve. What we need is a system that addresses two fold tasks: the recognition system should be designed such that it is impossible or infeasible to search a given fingerprint database with latent prints.

Now that we have introduced ForestFingers we can show how they can support id-privacy. To implement 2-id-privacy, we choose the end-points of the pairs used to define the pair table on separate fingers. Then to match the system will need at least two fingers to define any of the pairs used for recognition. That definition presumed we can clearly place data on different fingers, which suggests segmentation. However we can skip segmentation by simply ensuring the minimum distance between the two minutiae in a pair is greater than the maximum (or expected) distance between minutiae within a single finger. The latter is the approach used for the experiment herein, where we uses distances that span 2 or 3 fingers.

This approach of defining graphs of pairwise features, where each pair has elements distinct elements of the i different data sources, can clearly be applied a much broader set of algorithms, including across multi-modal biometrics and mixing biometric and non-biometric data. The pairs need not be graphical elements, as we used herein, but could also be condition pairs, where one elements defines a local transformation of the feature, similar in spirit to how [4] used a password to mix the data.

One of the elements that must be addressed is how to make the pair-features consistent and deal with noise. In the case of slaps the local image coordinate system and physical repeatability of close finger placements, provide that consistency. Some alternatives would be, if there was consistent segmentation and reliable key feature localization (e.g. core/delta), they could be used to align the sets of data to support pair features. And for larger i levels of id-privacy , triangle or more complex subsets can be defined.

In order to test our hypothesis about 2-id-privacy we performed some initial experiments on mixing data from multiple fingers. We term this approach as cross-finger matching, i.e. using information from minutia points from different fingers and using relative distances and angles to form forests of fingers. We would like to bring it to the notice of the reader, that this approach is first of its kind and is by no means meant to be authoritative. This study is meant to open doors in this direction, where data from multiple fingerprints captured simultaneously can be used for matching and researchers should consider such information when designing their own algorithms.

In analyzing the cross-finger recognition data, three things became apparent. First, quality was having a significant impact significantly increasing false matches and their scores. Second, there were considerably more high-scoring false matches than in the per-finger matching. Thirdly, there were some false reject that have particularly low-scores. We briefly discuss how these might be addressed.

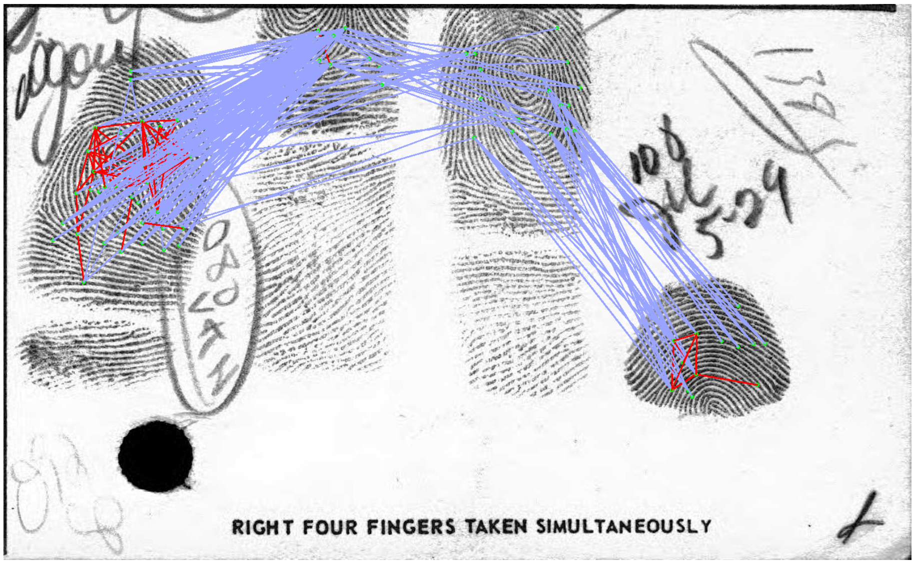


Figure 4.2: The above figure shows formation of forests of trees that span one and multiple fingers. This approach is designed to help the cross-finger recognition, but at the same time avoid matching with latents

4.4 One and Cross Finger Matching

With similar intentions in mind, we present another hypothesis: Mixing of data from single finger and multiple fingers in a way that mixing of data from single finger can only help in recognition and still avoid matching with latents. A thorough information theorectic analysis is needed to validate how much amount of data from single finger can be added so that to help recognition from cross fingers but still make it infeasible to match with latents.

4.5 Performance on latents

It is worthwhile to note the performance of cross fingers and one and cross fingers on latents. In order to simulate this, we considered a segmented finger from NIST DB29 slaps as a probe (we consisder only index finger for initial experiments) and slap images from NIST. Figure 4.3 summarizes the efforts in this direction. The data mixing approach makes search with latents impossible (rather "infeasible") at the same time maintaining accuracy for slap matches. We agree that the accuracy rate for matching slaps with data mixing approach is not usable in a real time system, but it definitely shows that we are proceeding in the right direction. We also found that many more edge pairs are formed when we are considering data from multiple fingers and hence only top 100 minutiae points were used for the above experiments. It should be noted that this is first of its kind approach and we expect people to build on top of this.

4.6 Effect of quality on matching algorithm

The quality issue is aggravated by the fact that this data was from scanned FBI fingerprint cards and not a live-scan device. To help understand the issued we reprocessed the matching using subsets of the data where we filtered on quality. We removed some extremely bad images, and the 2 duplicates in the dataset and the 2 people where the images were

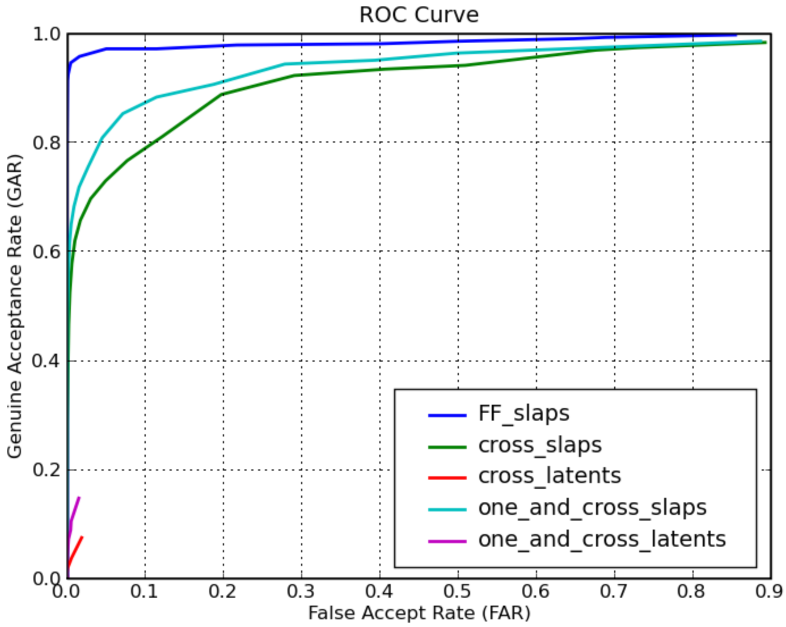


Figure 4.3: The above figure shows a systematic comparison of ForestFingers on straight slaps, and ForestFingers with data mixing enabled from multiple fingers. From the curves it is evident it is infeasible to search with latents with the given representation. The True positive rate does not increase

not actually slaps but where each inked finger was applied separately (including having duplicated fingers on the card). Then we processed for actual print quality. Unfortunately the NFIQ program does not seem to measure quality of an slap image properly, so we instead use NFIQ on each of the segmented images with the requirement that 3 of the four slap fingers have a quality of X or better. This reduces the population and number of matching attempts. For quality 2,3,4 and 5(all) the resulting subsets allowed for comparisons with Subjects/Total Matches of 40/12348, 129/52735,186/80914,209/90999 respectively. Figure 4.4 shows the ROC curves for cross-finger ForestFinger matching results for different levels of quality.

The high-scoring false matches could be reduced by adding more descriptive features to the minutiae. This paper uses basic features that would be computable with any ANSI/ISO-standard fingerprint minutiae extractor. It is well documented that other, often proprietary, features can improve per-finger matching algorithms. It is expected that the added features would more significantly improve cross-finger matching because of the birthday-paradox effect (N-squared potential pairs significantly increase chance of an unexpected match). These features may expose some information about the individual suitable for latent matching, so some care would be needed in their design/usage. To show the potential for this we added a small amount of local-neighborhood information by allowing pairs with very short distances to be part of the forest information. This

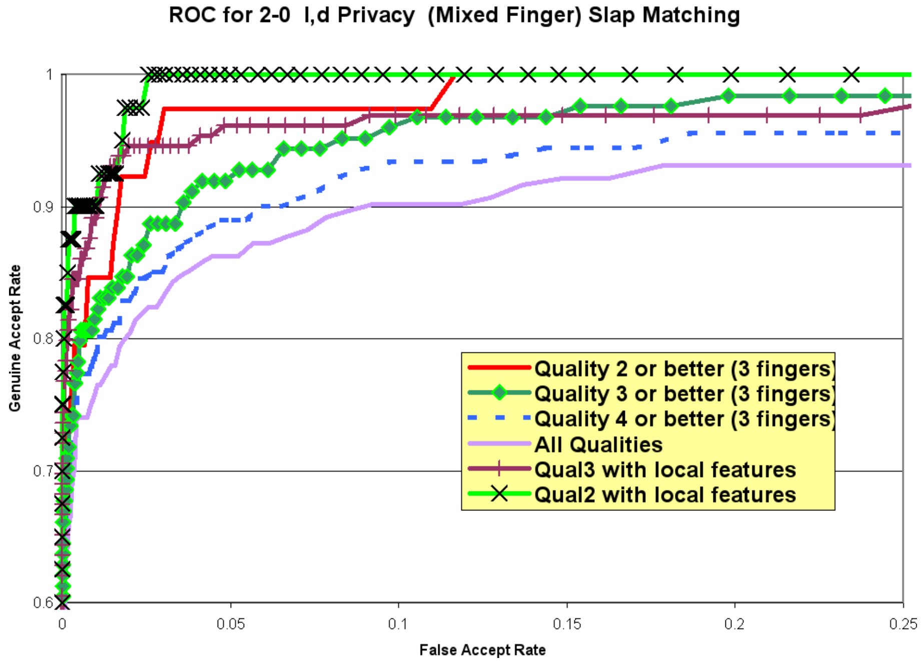


Figure 4.4: ROC curve showing performance of the 2-id-privacy mixed-finger approach using ForestFingers. Because of the weak definition of features for pairs, this is more significantly impacted by fingerprint quality. The curves shown are for different levels of minimum quality, with each curve using only data with at least 3 fingers per slap having the specified NFIQ quality or better. Also shown are cross-finger matching with one per-finger local pairs.

local per-finger information we no-longer have (2-0), as there is some potential for local matching (which could be determined empirically for a real dataset). We note that if one were to use the privacy enhancing transforms of [7], which can be applied to these types of “pairs”, the modulus operation of the transform would reduce the large and small distances into the same range and greatly increase the privacy element.

4.7 Study of Important Parameters

It is worth to mention what are the key parameters that are important for the results that we mentioned. First of all the total number of minutiae points allowed for recognition significantly affects the amount number of trees formed. Another important parameter was the length of allowable distance spacing between minutiae points. We also found that performance for forestfingers improved as the total number of availbale high quality minutiae points increases, when we were matching forestsfingers on unsegmented slaps. This is quite intuitive, since as more number of interest points are present in probe and gallery simultaneously, we can find significant amount of local structure across different fingers.

It is worth mentioning that some modifications like addition of “quality score” during pair-table and match-table formation and then pruning selectively based on these quality scores significantly affected the performance. The reason for this behavior is that formation of CMPG in forestfingers is a greedy process and high quality edge pairs if favored,

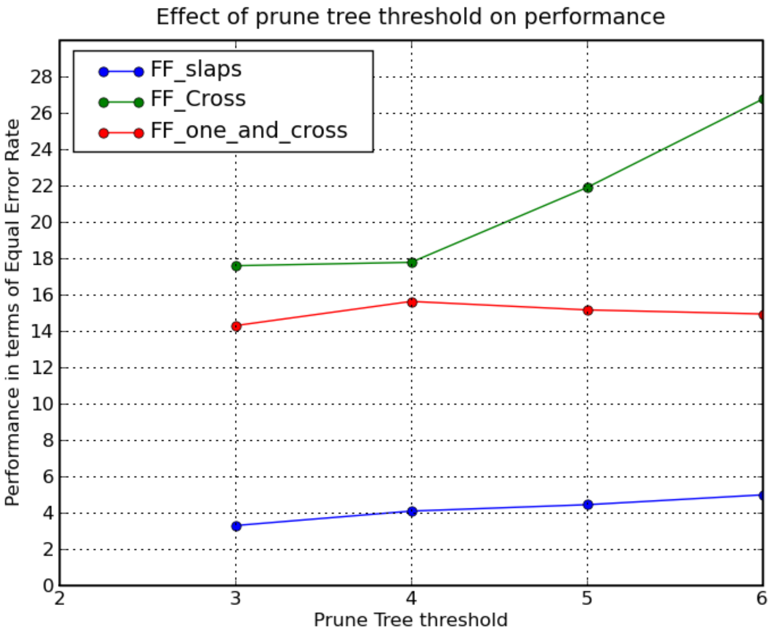


Figure 4.5: The above figure shows effect on performance as we change prune tree threshold. Performance is quantified in terms of equal error rate where smaller value mean better accuracy. From the above experiments, it seems that prune tree threshold of 3 seems to be optimal in terms of capturing enough local structure in case of true matches at the same time discarding formation of random structure in case on non-matches

gives consistent structure. In the final stage we consider only trees of significant sizes. As we increased the pruning threshold for these trees, we found that high-quality true matches weren't affected much but high scoring false matches weren't able to find consistent structures. However, because of low quality nature of dataset, increasing prune tree threshold also seemed to impact the overall performance. Fig 4.5 shows the effect on performance as a function of prune-tree threshold of the three methods discussed above.

Chapter 5

Conclusion

5.1 Contributions of the thesis

The contributions of this work are two fold:

1. This approach introduces a method for matching slap images without segmentation without loss in accuracy. Though the performance is far form being deployed in real world, the approach is promising and the work is very much in progress.

2. The second contribution of this work is that related to privacy. We introduced a method of mixing data from multiple fingers, which allows us duplicate detection but does not allow search of database based on latent prints. This idea would be key for preventing function creep in large-scale biometric programs

5.2 Conclusion and Future Directions

We showed how our approach solves 2-id-privacy, with ROC curves showing its accuracy. The performance is, admittedly, not yet as good as using per-finger features and not sufficient for large scale de-duplication. The goal here was to define a problem and a new model and show that has some potential. The early protected-templates research did not provide sufficient performance to be of use, but continued research increased performance and now there are multiple commercial products in that space.

To be effective, the approach needs to use more powerful features, to reduce false pair matching, and probably could benefit from improved optimization during the forest formation. We hope that this introduction will encourage biometric algorithm designers, and even object recognition designers and the gurus of graph-cuts to join us in trying to solve this important problem.

Bibliography

[1] *http://www.biometrics.gov/nstc* Date of last access, Nov 19, 2009

[2] *http://www.dhs.gov/xtrvlsec/crossingborders/* Date of last access, Nov 19, 2009

[3] E.Tabassi C.I. Wilson R.M.McCabe S.Janet C.I.Watson, M. D.Garris. *Nist fingerprint image software 2.* 2004

[4] A.Hicklin, *Slap Fingerprint Segmentation Evaluation*

[5] S.Prabhakar, S.Pankanti, A.K.Jain, *Biometric Recognition: Security and Privacy Concerns* IEEE Security and Privacy, vol. 1, no. 2, pp. 33-42, Mar. 2003,

[6] P. Tuyls, A.H.M.Akkermans, T.A.M.Kevenaar, G.J.Schrijen, A.M.Bazen, R.N.J.Veldhuis *Practical Biometric Authentication with Template Protection* AVBPA 2005

[7] T.E. Boult, W.J. Scheirer and R. Woodworth *Revocable Fingerprint Biotokens: Accuracy and Security Analysis* IEEE CVPR 2007

[8] N. Ratha, S. Chikkerur, J. Connell, and R. Bolle, “Generating Cancelable Fingerprint Templates,” *IEEE TPAMI*, vol. 29, no. 4, pp. 561–572, 2007.

[9] K. Nandakumar, A.K.Jain and S.Pankanti *Fingerprint-based Fuzzy Vault: Implementation and Performance* IEEE Transactions of Information Forensics and Security, Vol. 2, No. 4, pp. 744-757, Dec 2007

[4] K. Nandakumar, A. Nagar and A. K. Jain", "Hardening Fingerprint Fuzzy Vault Using Password," In *2007 Int. Conf. on Biometrics*, LNCS 4642, 2007.

[10] S. A. Cole, More Than Zero: Accounting for Error in Latent Fingerprint Identification *Journal of Criminal Law and Criminology*, Vol. 95, 2005

[11] Shamir, Adi (1979), "How to share a secret", *Communications of the ACM* 22 (11): 612613,

[12] E. M. Newton, L. Sweeney, B. Malin, "Preserving Privacy by De-Identifying Face Images," *IEEE Trans. on Knowledge and Data Eng.*, 17(2), pp. 232-243, 2005.

[13] D.Maio, D.Maltoni, S.Prabhakar and A.K.Jain *Handbook of Fingerprint Recognition*, Springer 2009

[14] X.Jiang, W. Yau *Fingerprint Minutiae Matching Based on the Local and Global Structures* vol. 2, pp. 10421045, ICPR 2000

[15] N.K.Ratha, V.D.Pandit, R.M.Bolle, V.Vaish *Robust Fingerprint Authentication Using Local Structural Similarity* WACV 2000

[16] C.I.Watson *NIST Special Database 29: Plain and Rolled Images from Paired Fingerprint Cards*

[17] B. Ulery, A. Hicklin, C. Watson, M. Indovina, K. Kwong, *NISTTIR 7209 Slap Fingerprint Segmentation Evaluation 2004 Analysis Report*, March 2005.

[18] R.Sedgewick *Algorithms in C* Addison-Wesley, 1997

[19] Rachael King "‘Homeland Insecurity” *http://www.businessweek.com* Date of last access Dec 16 2009