USER AUTHENTICATION USING ON-LINE SIGNATURE AND SPEECH

By

Stephen Krawczyk

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science and Engineering

2005

ABSTRACT

Ensuring the security of medical records is becoming an increasingly important problem as modern technology is integrated into existing medical services. As a consequence of the adoption of electronic medical records in the health care sector, it is becoming more and more common for a health professional to edit and view a patient's record using a tablet PC. In order to protect the patient's privacy, as required by governmental regulations in the United States, a secure authentication system to access patient records must be used. Biometric-based access is capable of providing the necessary security. On-line signature and voice modalities seem to be the most convenient for the users in such authentication systems because a tablet PC comes equipped with the associated sensors/hardware. This thesis analyzes the performance of combining the use of on-line signature and voice biometrics in order to perform robust user authentication. Signatures are verified using the dynamic programming technique of string matching. Voice is verified using a commercial, off the shelf, software development kit. In order to improve the authentication performance, we combine information from both on-line signature and voice biometrics. After suitable normalization of scores, fusion is performed at the matching score level. A prototype bimodal authentication system for accessing medical records has been designed and evaluated on a truly multimodal database of 100 users, resulting in an average equal error rate of 0.72%.

TABLE OF CONTENTS

Page

Chapters:

Appendices:

LIST OF FIGURES

LIST OF TABLES

Table **Page**

CHAPTER 1

Introduction

Every year, billions of patients in the United States visit doctor's offices, clinics, HMO's, hospitals, and other heath care providers [9]. Each of these visits either generates a new medical record or adds to an existing one, necessitating the retrieval of a particular record. Medical records contain extremely sensitive and personal information and should be stored under a high level of security in order to protect the privacy of the patient. In terms of what type of information is generally recorded in a patient's record, the regulations differ depending on the type of medical facility and the "accepted medical practice" [9]. This record documents the patient's history, physical findings, treatment, and course of disease. Health care facilities and hospitals have additional federal requirements that they need to conform to.

1.1 DNA Data

Another issue that should be brought to attention is that, in the future, our DNA sequences are likely to be included in our medical records. Deoxyribonucleic acid (DNA) is the carrier of genetic information in all cells and many viruses. The development of methods to isolate and clone large DNA fragments has led to much research pertaining to the human genome project; a project that intends to map and sequence the human genome. Facilities have already sequenced numerous basic genomes and a "rough draft" of the human genome with the finishing touches currently underway [18]. Knowledge of the genomes may reveal important information about disease, development, neurobiology, aging, and many other biological processes. While we do not yet completely understand how to interpret the code hidden in our genes, it is just a matter of time before the code is cracked. Once broken, holders of a sample of an individual's DNA will be able to learn more and more about that individual and his or her family. DNA may be considered unique and significantly more personal and private than any other information currently in our medical records. In the past, people have placed special emphasis on information that is potentially embarrassing and sensitive (such as sexually transmitted diseases) and uniquely personal (such as a photograph of one's face). Genetic information contains both these traits; it is both potentially embarrassing and uniquely personal. An individual is very sensitive to the disclosure of such private information because he or she could be discriminated, for example, by potential employers. Such a situation can be illustrated by considering a patient with symptoms for early onset of Alzheimer's disease. If a potential employer discovers this in the individual's medical record, this person's chances of getting the job might drastically decrease. A similar situation can be imagined when an individual is applying for life insurance.

No genetic employment discrimination case has yet been decided in either the U.S. federal or state court, however, the U.S. Equal Employment Opportunity Commission (EEOC) recently settled the first lawsuit alleging such discrimination [33]. In the lawsuit, the EEOC alleged that the Burlington Northern Sante Fe (BNSF) Railroad subjected its employees to blood testing for a genetic marker linked to carpal tunnel syndrome. BNSF was attempting to avoid the payment of compensation for the repetitive stress injuries that occurred frequently among its employees. At least one employee was threatened with discipline and possible termination for refusing to take the genetic test. The EEOC alleged that the tests were unlawful because they were not job related or consistent with any business necessity and soon reached a settlement with BNSF.

1.2 Government Regulations

The federal government has initiated regulations to protect the privacy of patients' medical data. Automation of health care information management has created increasing governmental and societal concerns about the security of computerized health care data. While the health care industry has incorporated electronic medical records, data repositories, networking, Internet access, and other new technologies into its various processes, the corresponding security measures have not been enhanced. Many weaknesses have been identified in existing health care security measures from past operations [6]. The Clinton Health Security Act proposed extensive federal rules regarding the collection and use of medical data. Although the plan itself was never fully accepted, many of its medical data provisions were used in the 1996 Health Insurance Portability and Accountability Act (HIPAA) that took effect April 2003. HIPAA included a mandate for standards that would ensure the security and integrity of health information that is maintained or transmitted electronically. The rules are numerous and complicated and most are beyond the scope of this work. Our work will take into account a subset of these regulations. The specific rule that is relevant here is that the patients are assured, under the HIPAA regulations, that their medical records will be used only by individuals directly involved in their medical treatment, payment of their bills, and health care operations. Any other individual or organization wishing to access the medical records would require specific authorization by the patient. This rule attempts to ensure that when medical records are properly disclosed; only the minimum amount of information necessary shall be accessed. This security rule focuses both on external and internal security threats. An example of a threat from an external source is someone posing as organization employees to access secure information. The internal threats are of equal concern and are far more likely to occur according to many security experts [16]. Facilities maintaining medical records must protect against careless staff or others who are unaware of security issues, and curious or malicious personnel who deliberately take advantage of the vulnerabilities of the system to access personal health information. It should also be noted that these rules are a minimal requirement and individual facilities can add additional security measures if desired.

1.3 Accessing Medical Records

Some patient records could span hundreds of pages. Accordingly, health care providers, in the interests of medical science and good patient care, recommend that medical records should be retained for as long as possible. These two factors suggest the move toward electronic records will greatly assist in the storage and management of the patients records. Consequently, since it is convenient for doctors to have a patient's record readily available when prescribing or administering treatment, many facilities have adopted the use of tablet PCs as access devices to retrieve and edit a patient's records on-line because of its ease of use. The widespread deployment of wireless access points presents the security problem of verifying whether the user of the tablet PC is authorized to view the requested private medical records. The level of security when accessing medical records on a tablet PC must be high enough to at least meet HIPAA's regulations. Patients' records are extremely private and especially with the widespread use of DNA analysis, they must be well protected from unauthorized users. While maintaining the convenience for doctors to be able to easily access medical records, it must be ensured that the individual using the tablet PC is authorized and, if that is the case, then only the minimum amount of information necessary is released based on his/her access privileges.

Figure 1.1: The use of a tablet PC by a medical professional.

1.4 Biometrics

Current mainstream authentication techniques to access restricted medical records use lock and key or password authorization. Many security problems arise when implementing such techniques to protect extremely confidential information. With passwords, most people set them to something they can easily remember such as family member's names, birthdays, famous athlete's names, etc. Passwords of this type can be easily broken by a brute force dictionary attack [41]. Because of this, many applications now require users to have passwords that must exceed a certain length, contain both characters and digits, and are not easily recognizable. A new problem arises with complicated passwords where users will write them down, since they can no longer easily remember them, possibly allowing an unauthorized user to locate and use the password. Another obvious security problem pertaining to passwords is that many people use the same password for multiple applications because of the difficulty of remembering multiple passwords and which password corresponds to which application. In such a situation, a breach in security in one application can compromise the security in many other applications. In lock and key authorization, the obvious problem is having the key or token shared, duplicated, lost, or stolen. The risks mentioned above are not acceptable for medical records; an alternate authentication protocol must be determined.

Biometric authentication alleviates many of the risks associated with lock and key or password authorization. Biometric authentication can be defined as automatic recognition of a person based on his or her physiological or behavioral characteristics [35]. To recognize an individual, biometrics relies on who you are or what you do instead of what you know, such as a password, or what you possess, such as an ID card. Biometric systems run in either identification mode or verification mode. During identification, the system recognizes an individual by comparing the captured biometric characteristic with all the biometric templates that are stored in a database for each user. This process will fail if the individual is not enrolled in the system. This is referred to as a one-to-many matching because the system must compare the input with every template stored in the database. During verification, the system recognizes an individual by comparing the input characteristics with the individual's biometric template. This is known as a one-to-one matching since it is known which user is attempting to be authenticated. Biometric verification is positive recognition, the same type of recognition used by lock and key and password authentication. The main difference is that biometrics cannot be lost or forgotten and on-line biometrics systems require the individual that needs to be authenticated to be physically present.

1.5 Multibiometrics

Having presented that biometric systems have a security edge over traditional methods in that they cannot be easily stolen or shared, it should be mentioned that these systems also have their own limitations. Common problems that may occur in biometric systems are

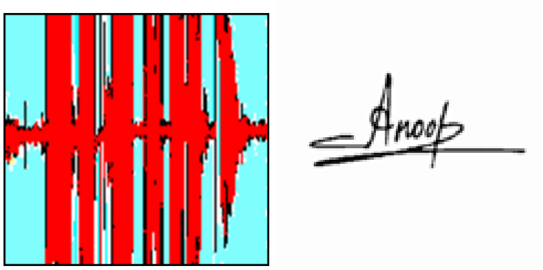


Figure 1.2: Visualizations of voice and signature biometrics.

noise in sensed data, intra-class variations, distinctiveness, and spoof attacks [24]. Noise in the sensed data can be caused from noisy inputs (such as a person with a cold speaking into a microphone), improperly maintained sensors (such as a defective pen for a tablet PC), and unfavorable ambient conditions (such as a person speaking into a microphone in a noisy room). Noise in the biometric data may cause a genuine user to be rejected by the system or an impostor to be recognized as a valid user. Intra-class variations occur when the biometric data acquired from a person during authentication is very different from the data used to generate the biometric template during enrollment. These variations will greatly affect the matching scores produced. This type of error may occur for a user who signs his or her name in multiple styles. The problem of distinctiveness occurs when the expectation of biometric traits to vary significantly between individuals is not met. There may be large similarities in the feature sets used to represent the biometric traits of two different individuals. Finally, spoof attacks occur when an impostor attempts to spoof the biometric trait of an authentic, enrolled user in order to gain access to the system. The biometric security problems presented assume that a unimodal biometric system is being used; i.e. a biometric system that relies on the evidence of a single biometric trait. Many of these limitations imposed by unimodal biometric systems can be either overcome or reduced by using multiple biometric traits. These systems can expect more accuracy due to the fact that they use multiple biometric modalities where each modality presents independent evidence to make a more informed decision. If any of the mentioned limitations is present in one extracted biometric trait, there will be other traits for the system to use in its decision. In terms of spoofing attacks, it becomes much more difficult for an intruder to simultaneously spoof the multiple biometric traits of a legitimate user.

1.5.1 Biometric Fusion

When using multiple biometric traits in an authentication system, it is necessary to determine a method by which the individual modalities are combined. This is referred to as biometric fusion and can be performed at three possible levels: fusion at the feature level, fusion at the matching score level, and fusion at the decision level. Fusion at the feature level is expected to perform the best but it is not always feasible. One problem is that most commercial systems do not provide information at this level. Also, the feature spaces of different biometric traits may not be compatible. Additionally, concatenation may result in a feature vector with a very large dimensionality leading to the "curse of dimensionality" [40]. Because of these limitations, we will use fusion at the matching score level. Three matching score level fusion techniques have been investigated by Ross and Jain [23]. The first is the sum rule where the weighted average of the scores from the multiple modalities is used to make the final decision. A second method uses a decision tree where a sequence of if-then-else rules are derived using the training set in order to assign a class label to the input data. The third approach is to use a linear discriminant function where, first, the score vector is transformed into a new subspace that maximizes the between class separation. The test set vectors are then classified by using the minimum Mahalanobis distance rule. In their tests, the sum rule provided the best performance.

When using this rule, user specific weights can be incorporated to improve the accuracy of the biometric system. Different users tend to adopt differently to individual biometric indicators; accordingly, if one trait is more consistent (lower intra-class variability) than the others, it should be given more weight. These weights can be discovered over time by examining the stored template of the user, the query set provided by the user, and the matching scores for each of the individual modalities. Such a system should provide the security and convenience desired for our application.

1.6 Proposed Solution

As mentioned earlier, user names and passwords are not only insufficient for security reasons but also inconvenient for the user of a tablet PC. This is because the tablet PCs normally use a stylus or pen for user input instead of a keyboard. Input can be conveniently obtained from the user either using the pen or the microphone that all tablets are built with. The types of biometric information that can be gathered unobtrusively from these devices are a person's signature and voice. Our solution to this problem will use both of these biometrics in order to construct a multimodal authentication system for accessing the patient records.

Several challenges arise when attempting to build a system that uses these biometrics for user authentication. The first is speaker identification; the system must find out which doctor or patient is trying to log into the system by what the user is saying. The voice biometric is more suited for identification compared to signature since a person's signature is sometimes very difficult to associate with the actual name. Sometimes, it actually need not pertain to the person's name in any apparent way. Also, the error rates associated with the signature trait are generally higher than those of the voice biometric. Rather, we will use the spoken name of the user and attempt identification based on the audio signal captured using the microphone. The next two challenges are of verification; using both speech and signature. Speaker verification is different from speaker identification in that it is not trying to find a speaker out of a group of speakers but trying to use the characteristics of the voice to make sure that the user matches with the template of the claimed identity. Signature verification has a similar challenge where the characteristics of the digital ink input by the user are examined to see if the signature is genuine. The final challenge is to provide a sufficient level of security so that it is extremely unlikely that an unauthorized user is able to log in and, at the same time, make it very likely that a genuine user is able to access the system.

1.7 Performance Evaluation

The performance of biometric verification systems is typically described in terms of the false accept rate (FAR) and a corresponding false reject rate (FRR). A false acceptance occurs when the biometric system allows an impostor to access the system. A false reject occurs when a valid user is rejected from gaining access to the system. These two errors are directly correlated, where a change in one of the rates will inversely affect the other. A common alternative to describe the performance of system is to provide the equal error rate (EER). This value corresponds to the point where the false accept and false reject rates are equal. In order to visually depict the performance of a biometric system, receiver operating characteristic (ROC) curves are drawn. Biometric systems generate matching scores that represent how similar (or dissimilar) the input is compared to the stored template. This score is compared to a threshold to make the decision of rejecting or accepting the user. This threshold can be changed in order to obtain various FAR/FRR combinations as shown in Figure 1.3.

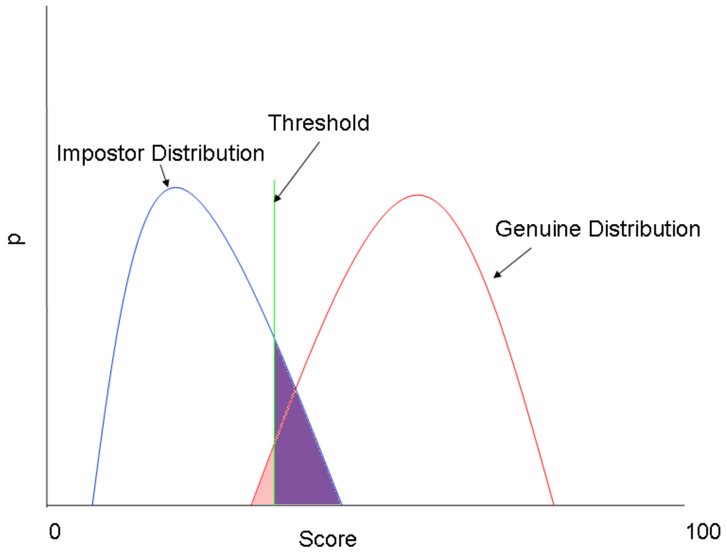


Figure 1.3: Plot of the genuine an impostor distributions and a decision threshold. The genuine and impostor distributions are labeled along with a threshold. False acceptances are impostor users who are accepted by the system and these occurrences are darkly shaded. False rejections are genuine users who will be rejected by the system and these occurrences are lightly shaded.

The ROC curve displays how the FAR changes with respect to the FRR and vice-versa. An example ROC curve is shown in Figure 1.4. These curves can also be plotted using the genuine accept rate versus the false accept rate. The genuine accept rate is simply one minus the FRR.

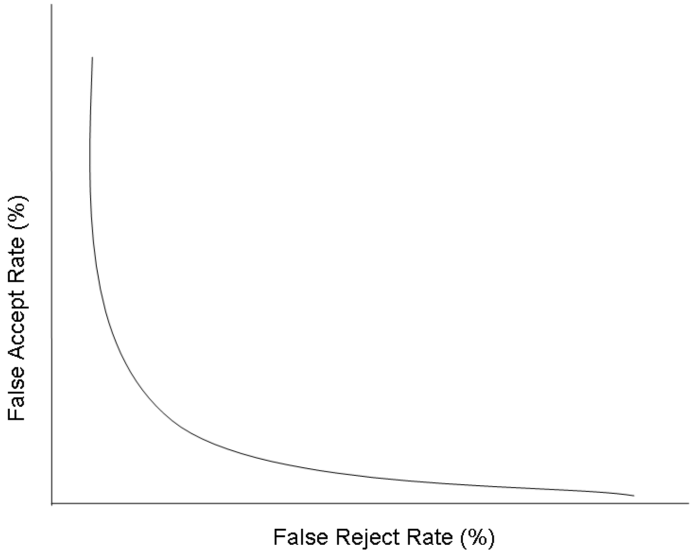


Figure 1.4: Example of a receiver operating characteristic (ROC) curve.

1.8 System Design

A schematic diagram of the multimodal system is presented in Figure 1.5. A user, whether it is a doctor or patient, trying to gain access to patient records will first have to speak his or her name. Next, if the name is correctly recognized by the system, the user will enter his or her signature. The system will uses the characteristics of both the voice and the signature to decide if the user is authorized to access the system. The above situation assumes that the user is already enrolled in the system. Enrollment requires the user to input several signatures and speak their name multiple times in order to create a template for that user. This template, containing the description of the user's voice and signature, is stored in the system's database and later used for matching. During matching, the first task the system needs to perform is speaker identification; i.e., determine which user is speaking. This can be done using a simple speech to text recognition engine and then comparing the resulting text with the stored user names [25]. We use a more robust technique that acoustically analyzes the voice signal to perform identification and generates a list of the potential identities. This is done so that the verification process is a one to one matching as opposed to a one to many matching. If it is known exactly which user is trying to log into the system, only one template needs to be matched with the user input. This not only greatly increases the speed of the authentication process, because the matching algorithms are responsible for the bulk of the processing time, but also the level of security. If the identification generates at least one potential identity enrolled in the system, the verification process will begin. Voice verification tries to match the characteristics of the input from the microphone with the stored voice template of the user [11]. Most matching algorithms output a score that represents how close the input is to the template. Verification is performed on each of the potential identities and the identity with the highest score is used. If this score is greater than some preset threshold then the voice can be accepted as genuine. The signature verification works in a very similar fashion where the digital ink is used as input to match with the stored signature template of the identity generated by the voice recognition system and a matching score is produced [21]. Using both voice and signature verification will reduce the probability of unauthorized access while maintaining a desired genuine accept rate. The system will be able to handle variations in the data by analyzing both the biometric modalities to obtain a decision. A method for combining the matching scores of the voice and signature is explored and the result is compared against a threshold in order to decide if the input is genuine. This threshold can be adjusted such that the performance of the system will meet the requirements of the domain.

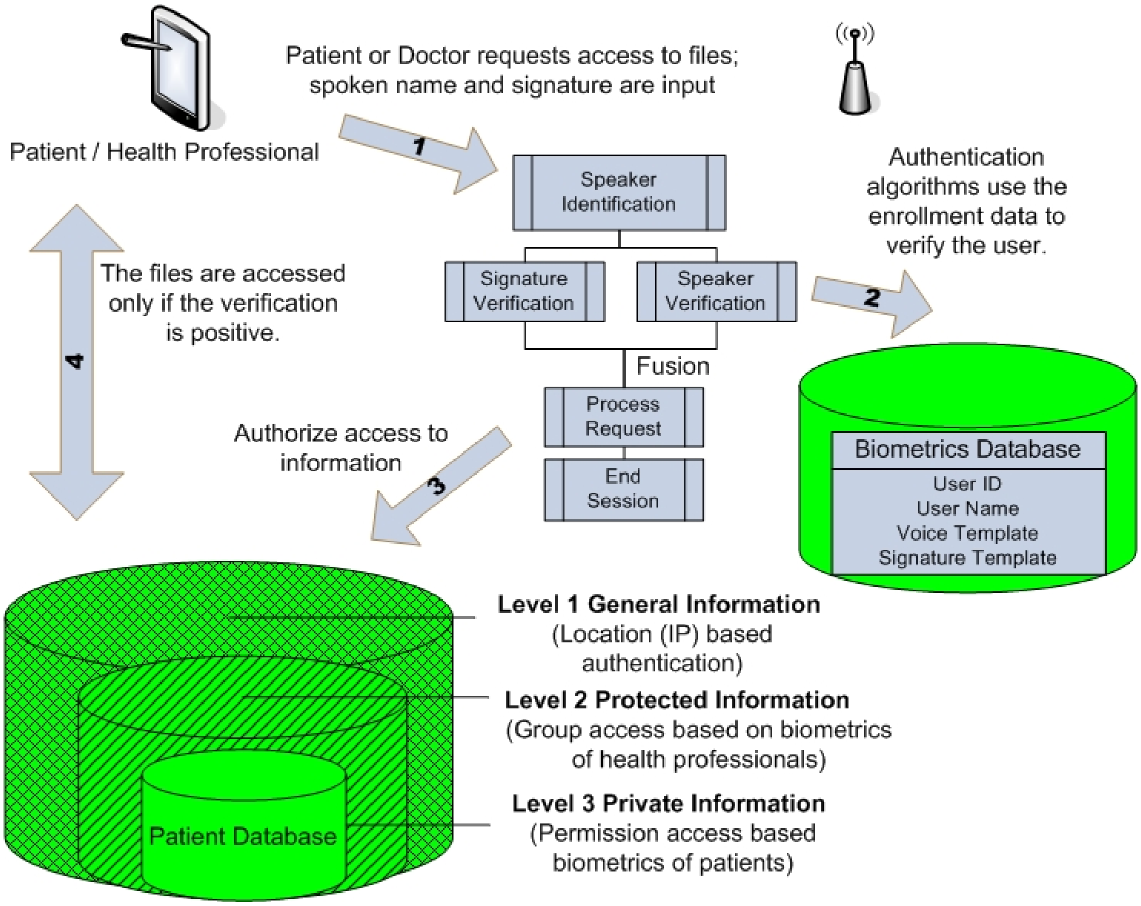

Patient or Doctor requests access to files;
spoken name and signature are input
Patient / Health Professional
1
Speaker Identification
Signature Verification
Speaker Verification
Fusion
Process Request
End Session
Authentication algorithms use the enrollment data to verify the user.
2
Biometrics Database
User ID
User Name
Voice Template
Signature Template
The files are accessed only if the verification is positive.
4
Authorize access to information
3
Patient Database
Level 1 General Information
(Location (IP) based authentication)
Level 2 Protected Information
(Group access based on biometrics of health professionals)
Level 3 Private Information
(Permission access based biometrics of patients)

Figure 1.5: Proposed system for tablet PC multimodal authentication.

The database of medical records is partitioned based on access privileges. The billing department does not need to view a patient's entire medical record but only the parts regarding the patient's current address, insurance information, etc. This type of information is included in the top partition. The second partition is for use of medical professionals and contains information regarding diseases, allergies, medical history, prescriptions, etc. The final partition of the database stores information most private to the patient and will only be released to very few individuals that require access to this information. Accordingly, only the minimum amount of information is revealed to the individual wishing to access the records.

1.9 Past Work

As previously mentioned, the HIPAA rules were initially introduced in 1996 and included provisions that required the U.S. Department of Health and Human Services (HHS) to adopt national standards for electronic health care transactions. In 2001, as one of the last acts of the Clinton Administration, HHS issued “final” regulations entitled ”Standards for Privacy of Individually Identifiable Health Information.” This sparked much protest among the health care industry and, in particular, the health insurance companies who caused the Bush Administration to agree that the regulations were defective and opened them up for public comment again. Many of the deficiencies had to do with the complexity of the rules and the financial burdens that it put on the health care industry but they also included a key deficiency where biometrics was treated as a threat to medical security. Some of the rules prevented the use of biometrics because of the possibility of de-identification and re-identification of protected health information. It was thought that biometric templates could be used to reconstruct a person’s identity and thus should not be used to protect medical records. The International Biometric Industry Association (IBIA) added its public comment, asking the Bush Administration to treat biometrics as a strong means to secure medical privacy. The charter explained that the biometric templates normally use proprietary and carefully guarded algorithms to secure a record and protect it from disclosure. Having access to the template alone is of no use since it cannot be reconstructed to reveal a person’s identity and thus does not meet the definition of a “personal identifier” [5]. The requested changes were adopted, allowing biometrics to be used as an effective security tool to protect the medical records of patients.

1.9.1 Biometric Systems in Practice

Before the HIPAA rules were finalized, biometrics was already being used to allow patients to access their medical records, authenticate the identity of patients to reduce medical treatment errors, and prevent unauthorized personnel from accessing a patient's record. After the changes were made to the HIPAA rules as a result of the public comments and the compliance date of April 2003 was set, the appeal of biometrics as a solution to the security of medical records was more widely recognized. Many facilities such as A4 Health Systems [1], Blue Cross & Blue Shield of Rhode Island [2], Sharp Health care [7], Inova Fairfax Hospital [4], and Health Highway [3] have decided to replace insecure passwords with fingerprint authorization. Others, such as the University of South Alabama Hospitals [8], have switched to iris technology to grant permission to clinicians with appropriate access to view protected information and associated reports. The biometric systems being used in the medical industry are primarily unimodal. The fingerprint biometric seems to be getting the most attention with companies such as SecuGen, Bioscrypt, Identix, and SAFLINK integrating their solutions with the medical facilities mentioned above. Iris and signature biometrics are also being utilized in some rare occurrences.

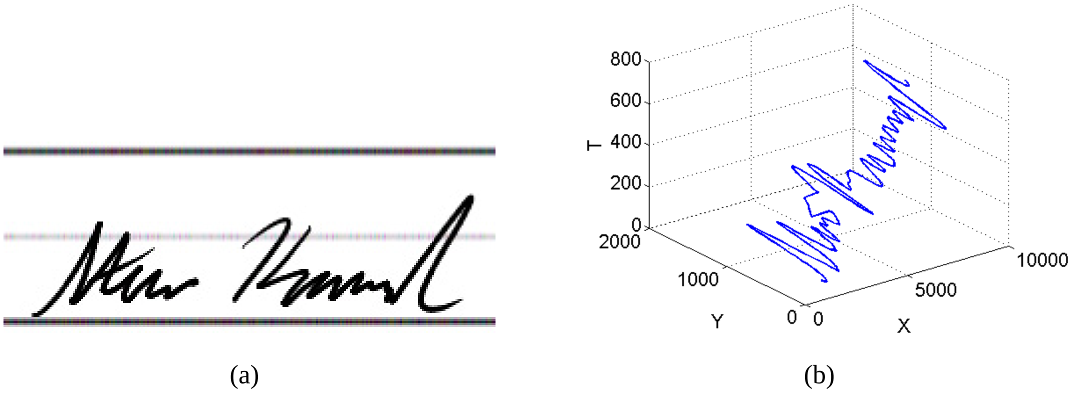
CHAPTER 2

On-line Signature Verification

Signature verification is a behavioral biometric that is developed over the course of a person's lifetime. Many people are very accustomed to the process of signing their name and having it matched for authentication. This process has been in practice for centuries and is well accepted among the general public to protect confidential information. The use of signature is prevalent in the legal, banking, and commercial domains.

Depending on the aquisition process, automatic signature verification systems are divided into two catagories; on-line and off-line. In off-line systems, a signature written on a piece of paper is captured optically with a camera or scanner. In on-line systems, the data is captured while the signature is being written. Aquisition is this form requires a special pen or digitizing tablet. These devices are able to capture both the static and dynamic attributes of a signature. Static attributes are the visible properties of the signature (e.g. shape, size, position) while dynamic attributes are the invisible properties (e.g. timing, pressure, speed). The invisible information gathered by on-line signatures makes them more reliable because timing and pressure attributes are much harder to imitate than the static information of a signature. Consequently, on-lines signatures are the focus of this chapter. Examples of both on-line and off-line signatures are displayed in Figure 2.1.

Work on automating the process of signature verification has been ongoing since the 1970's. In the following section, we highlight some of the major contributions to the area and attempt to provide an encompassing view of the various approaches used to solve the problems of automatic signature verification. A signature verification system must provide a solution to the problems of preprocessing, feature extraction, matching, and performance



(a) (b)

Figure 2.1: Examples of an off-line (a) and an on-line (b) signature. The on-line signature is displayed in an x versus y versus time plot.

evaluation. A diagram of a generic signature verification system is shown in Figure 2.2. Training signatures are provided to create a template for a user. These signatures are preprocessed and features are extracted and then stored. When a template is entered into the system with a claimed identity, the signature is preprocessed and the features are extracted. The features of the template signatures for the claimed identity are then matched with the features of the input signature. Finally, a matching score is produced.

2.1 Literature Review

One of the first published works on on-line signature verification was performed by Liu et al. [19]. In this work, handwriting was modeled as ballistic motions that do not involve sensory feedback. An example of another action that this model fits is the rapid saccadic motion of the eye, which consists of small jumps of 10-30 milliseconds in duration. Using this model, they conclude that forces are produced strictly in terms of magnitude and duration. Accordingly, a pen was designed that measures the muscle forces in the hand

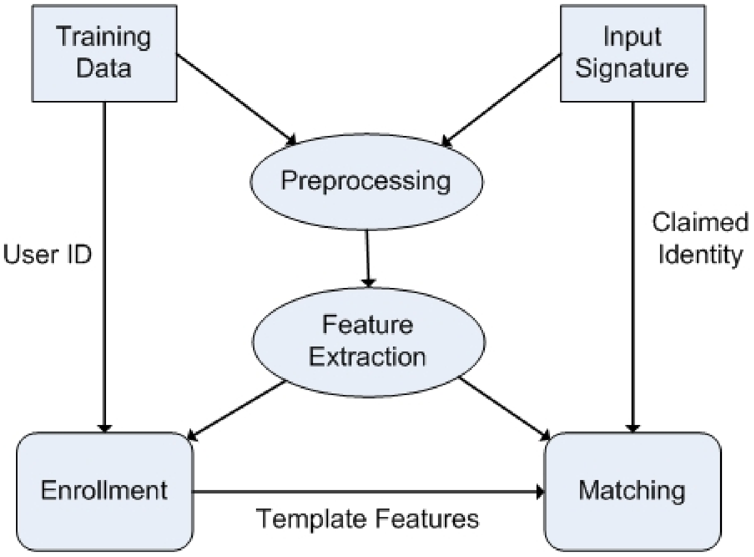


Figure 2.2: A typical signature verification system.

by recording the acceleration signal over time. In this study, it was noticed that the time interval for signing one's name is remarkably consistent.

To perform verification, the input and template signatures are divided into segments where the divisions are the points between strokes[1]. A stroke was further divided if it consisted of more than two seconds in duration. Next, cross correlation is performed using the acceleration signals for each segment. A final matching score is computed by summing the segment correlations, where each is weighted by 1 divided by the number of points in the segment. A form of majority voting was used when a input signature was compared against a set of templates in order to make a decision. A total of 1332 genuine signatures from 70 subjects was collected along with 287 skilled forgeries to be used as the database. The best results obtained by this system was a FRR of 2.9% with a corresponding FAR of 2.1%.

[1] A stoke is defined as the points between a pen down and a pen up.

A subsequent work by Liu et al. [31] used the same regional correlation algorithm as before but also experimented with the use of pressure information. It is argued that the full pressure signals provide little discriminatory information between subjects. In order to extract the most discriminative information from these signals, only the minutiae of the pressure signals are used. In this context, minutiae are calculated by subtracting the average pressure waveform from each segment and keeping only the points of zero-crossings. The pressure waveform consequently becomes 1 dimensional as a function of time (only the times of each zero crossing are stored). Cross correlation is again used to compare two pressure waveforms and this result is averaged with the correlation value from the acceleration signals. Tests were performed with 201 subjects and included more than 6000 signatures. Ten skilled forgeries[2] were collected for 40 of the users. Using a simple form of template updating, results of 1.7% FRR and 0.4%FAR were reported for random forgeries[3]. Only two out of the 400 skilled forgeries were falsely accepted, while no genuine signature were falsly rejected.

Following many achievements in the area of automatic signature verification and writer identification, Plamondon et al. presented a survey of the current state of the art in 1989 [34]. This work identified the five problems that need to be addressed in order to create a signature verification system: data acquisition, preprocessing, feature extraction, comparison process, and performance evaluation. By analyzing the literature according to the solutions to these problems, the authors characterized the current verification methods into two classes: functional and parametric. In functional approaches complete signals (x(t), y(t), v(t), etc.) are used directly or indirectly to compose the feature set. The challenges with this approach is during the matching process where two signals that likely have various

[2] Skilled forgeries are signature attempts by an impostor where the impostor is able to see how the genuine signature is signed before attempting the forgery.

[3] Random forgeries are genuine signatures taken from one user and used as input for another user.

durations and undergo non-linear distortions have to be compared. The advantages are evident during the feature selection process because the signals themselves or their derivatives are used as the features. The other class of verification methods is the parametric approach. These methods extract a fixed number of parameters from the complete signals. The difficulty with this approach is selecting the salient parameters that can distinguish between subjects and are consistent among the same subjects. However, once the parameters are selected, the matching of the parameters can be done using a variety of simple techniques. The authors conclude that the functional approaches are normally more computationally expensive and thus slower but they also provide higher accuracy.

Hidden Markov Models (HMM) are extremely powerful in speech recognition and have also been used in signature verification. Yang et al. trained HMMs to model the sequence of normalized angles along the trajectory of a signature [42]. The normalized angles were computed by extracting the sequence of absolute angles along the points of the signature and subtracting the starting angle from each absolute angle. This calculation is used to make the features rotational invariant. Also, size normalization is performed by uniformly dividing the signature into K segments, where K is the observation length for input to the HMMs. The actual features that are used as input for a given segment k, consisting of n samples is calculated using the equation

$$\phi(k) = \arctan\left[\frac{\sum_{l=i+1}^{i+n} s_l^{(k)} \sin \overline{\theta}_l^{(k)}}{\sum_{l=i+1}^{i+n} s_l^{(k)} \cos \overline{\theta}_l^{(k)}}\right], \tag{2.1}$$

where s_l^k is the distance between samples l and k and $\overline{\theta}_l^{(k)}$ is the relative angle between samples l and k. Next, this angle is quantized into sixteen discrete symbols. The samples of when the pen is up and when the pen is down are used independently and, accordingly, there are sixteen symbols for the angles when the pen is down and also sixteen different symbols when the pen is up. The path of the pen up samples is calculated by interpolating between pen downs. A form of user dependent thresholding is used, in which the threshold depends on the output of the models during training. The system was tested on a database of 496 signatures from 31 signers. Eight signatures from a subject were used for training the model and the other 8 were used for testing. In their experiments, only random forgeries were considered. Several parameters such as the number of states, the observation length, and various architectures for the HMMs were explored. For the individual HMMs, the Bauman Welch algorithm was used for both training and classification. Their best result exhibited a FAR of 6.45% and a corresponding FRR of 1.18%.

Many problems in pattern recognition have used integral transforms to perform feature extraction (e.g., Hadamard, Hough, Walsh, etc.). This type of feature extraction has also been attempted for signature verification. A successful signature verification system using an integral transform was developed by Lam and Kamins [29]. To extract features from signatures, the Fast Fourier transform (FFT) is performed and the most distinguishing harmonics are used as the features. These features are selected individually for each user, so that the most distinguishing harmonics are used pertaining to each specific user. Several preprocessing steps were performed before the transform is computed to normalize for duration, rotation, position, and size. Also, any stokes whose duration is less than a threshold are removed to prevent spikes in the transform. Next, the stroke segments are concatenated together to form one continuous signature, where even the beginning and end points are connected through linear interpolation (out of the 1024 points in the normalized signature, the last 25 points are used for this interpolation). The signatures are also normalized for drift, which the authors define as how the signature moves from one position to another (e.g., from left to right in the western world). Linear regression is used to eliminate the drift so that the end points of the signature do not excessively influence the transform. After preprocessing, the x and y time functions are transformed into the frequency domain and a cutoff is applied to remove the unnecessary high and low frequencies. Next, the top 15 harmonics with the highest magnitudes, normalized by the sample variances, are selected as the feature set. This algorithm was tested on a dataset of 20 signers, each contributing 8 genuine signatures and 19 other signers provided skilled forgeries. Using the leave one out method, the system generated an equal error rate of 2.5%.

Neural networks can solve complex functions by attempting to learn what the correct output should be from training data. In the past, neural networks have been successfully applied to handwritten character recognition and phoneme recognition. Lee attempted to use various neural network algorithms to classify a signature as either genuine or impostor [30]. He examined three neural network based approaches: Bayes multilayer perceptrons (BMP), time delay neural networks (TDNN), and input oriented neural networks (IONN). Preprocessing steps such as linear time normalization and signal resampling were performed. The input to the neural networks was a sequence of instantaneous absolute velocities extracted from the spatial coordinate time functions (x and y signals). This absolute velocity was computed for each point as:

$$\|v(n)\| = \frac{\sqrt{\Delta x(n)^2 + \Delta y(n)^2}}{\Delta t(n)}, \tag{2.2}$$

where $\Delta x(n)$, $\Delta y(n)$, and $\Delta t(n)$ are the change in x, y, and time values at point n, respectively. The problem with using neural networks for signature verification is that examples of forgeries are required to train the network for a user. The networks cannot be properly trained by being given only genuine samples. Accordingly, the database used consisted of 1000 genuine signatures from only one subject and 450 skilled forgeries from 18 trained forgers. The back propagation algorithm was used for network training. This experiment misclassified 2.67% of the input (it is not mentioned how many of the misclassifications were false acceptances and how many were false rejections).

On-line signatures may contain pressure, altitude, and azimuth information of the pen along with the x and y information. The x,y, and pressure signals have shown to remain consistent among the same user and also provide discriminatory information between separate users. The altitude and azimuth signals have not been as widely used partially because they cannot be acquired in many devices and also because they have not been shown to provide much discriminatory information between subjects. Hangai et al. implemented a system that compared the pressure, altitude, and azimuth signals [17]. In their work, the azimuth is referred to as the direction of the pen. The pen altitude, referred to as $\phi(t)$, and the direction, referred to as $\theta(t)$, are combined to form a three-dimensional feature vector in the following fashion:

$$v(t) = \left[\begin{array}{l} \sin\theta(t)\cos\phi(t) \\ -\cos\theta(t)\cos\phi(t) \\ \sin\phi(t) \end{array} \right] \tag{2.3}$$

Preprocessing steps, such as size normalization and alignment of the starting points of the two signatures are performed before matching. The authors use dynamic time warping to align the two signals. The database used for the experiments was gathered from 24 people who each contributed 25 signatures, where 5 were used for training and the other 20 for testing. It is also reported that skilled forgeries were generated for each user but the authors do not specify the size of the skilled forgery data set. Using only the derived altitude/azimuth feature vector, $v(t)$, yielded an accuracy of 93.3%. With pressure information alone, an accuracy of 87.8% was obtained. Finally, using the x and y information alone produced an accuracy of 85.8%. When all the information was combined, every signature was correctly classified. The results show that if the altitude and azimuth information is used in a correct fashion, it can sometimes provide as much or even more discriminatory information than either the pressure or shape information.

Dynamic time warping (DTW), a string matching technique, has received much attention in signature verification. The two main drawbacks of using this technique is that it has a heavy computational load and forgeries are warped to more closely match the genuine signatures. Feng and Wah attempted to resolve these issues by warping only what they call extreme points [13]. Extreme points are defined as signal peaks and valleys. Specifically, a peak or valley is marked as an extreme point if $r \geq h_0$ and $d \geq h_0$, where r is the rise distance (amplitude from a valley to the following peak), d is the drop distance (amplitude from a peak to the following valley), and h_0 is a user defined threshold. Parameter h_0 is used to eliminate small ripples in the signal; this is important because small ripples in the signal are normally unreliable. The matching of two strings of extreme points uses DTW but is modified so that only peaks (valleys) are matched to peaks (valleys). In their experiments, only the x and y features are used to determine the distance between two extreme points, where city-block distance is used as the metric. After the alignment is found between two sets of extreme points, the input signature is warped to match the reference signature. This is done by linearly stretching the points in the x direction, while keeping the y values constant. The final matching score is computed by calculating the correlation coefficient between the reference and the warped input signature. This algorithm was tested on a database of 25 users with 30 genuine signatures and 10 skilled forgeries for each user. Tests on this technique versus DTW is performed with equal error rates of 25.4% and 33.0%, respectively. Also the computation time for DTW was 11 times more than using the extreme points technique.

2.1.1 Human Signature Verification

It is also interesting to compare how automatic signature verification methods compare to performance of humans on the same task. Kam et al. reported results on the first controlled study comparing the abilities of forensic document examiners (FDEs) and laypersons in the area of signature verification [26]. A comparison of six known signatures (genuine) with six unknown signatures (genuine and skilled forgeries) was performed by each subject to classify the six unknown signatures as either genuine or impostor. The subjects included 25 forensic document examiners and 50 laypersons where laypersons were provided with various monetary incentives. In this work, a forensic document examiner is considered to be a person that must satisfy one of the following:

- certified by the American Board of Forensic Document Examiners,
- member of the American Society of Questioned Document Examiners,
- member of the Southwestern Association of Forensic Document Examiners,
- or member of the questioned document section of the Mid-Atlantic Association of Forensic Scientists.

The results found that FDEs had a FAR of 0.49% and a corresponding FRR of 7.05%. The error rates for the laypersons were a FAR of 6.47% and a FRR of 26.1%. Since random forgeries are very easy for humans to distinguish they were not included in the testing. These results highlight that automatic signature verification systems rival and even surpass the abilities of forensic document examiners in terms of classifying skilled forgeries. The area in which the automatic signature verification systems are lacking is classifying random forgeries; humans have error rates very close to 0%, while for computers, this accuracy has not yet been reached.

2.2 Past Work

All of our work regarding signature verification was built on top of the earlier work performed by Jain, Griess, and Connell [21]. The following sections will give an overview of the signature verification algorithm that was developed in order to clarify contributions made in this thesis.

2.2.1 Preprocessing

In order to eliminate noise from the input signature, whether created by the digitization process of the input device, the speed of the writing, or the writing itself, the signature is smoothed by a Gaussian filter. A one-dimensional filter is applied separately to the x and y directions. The points in the signature are also resampled. This is because if two signatures are going to be compared with respect to their shape, they must be resampled in order to extract more reliable shape features. On-line signatures contain both spatial and temporal data. After resampling the signature, the temporal data will be lost because the spacing of the points represents the velocity with which they were written. Digitizing tablets sample at a constant rate, and accordingly, the sampling rate provides a uniform time unit. Thus, the only information needed to extract the speed of writing is to measure the distance between two consecutive points. To retain this information, temporal features are extracted before uniform resampling is performed. The resampling process in performed by creating equidistant spacing between each point, where the spacing distance is a user defined parameter. Finally, the authors define some points in the signature as critical points. These points carry important information about the structure of the signature and should not be changed throughout the resampling process. Specifically, critical points are endpoints of strokes and points of trajectory change.

Figure 2.3 displays some instances of critical points. The upper row displays critical points where the x or y direction of the stroke changes. The lower row shows critical points where a transition from a vertical or horizontal stroke forms in to a curve. These points are found before any other preprocessing steps are performed and the speed of writing at each of these points is stored. During resampling and smoothing, these points are not changed, ensuring that the important structural aspects of the signature are retained. The last step in preprocessing is to perform stroke concatenation. The strokes are combined into one long stroke in order facilitate the string matching process. Overall, the preprocessing steps are performed in the following order:

- Extract critical points
- Fine resampling
- One-dimensional Gaussian filtering in the x and y directions
- Coarse resampling
- Stroke concatenation

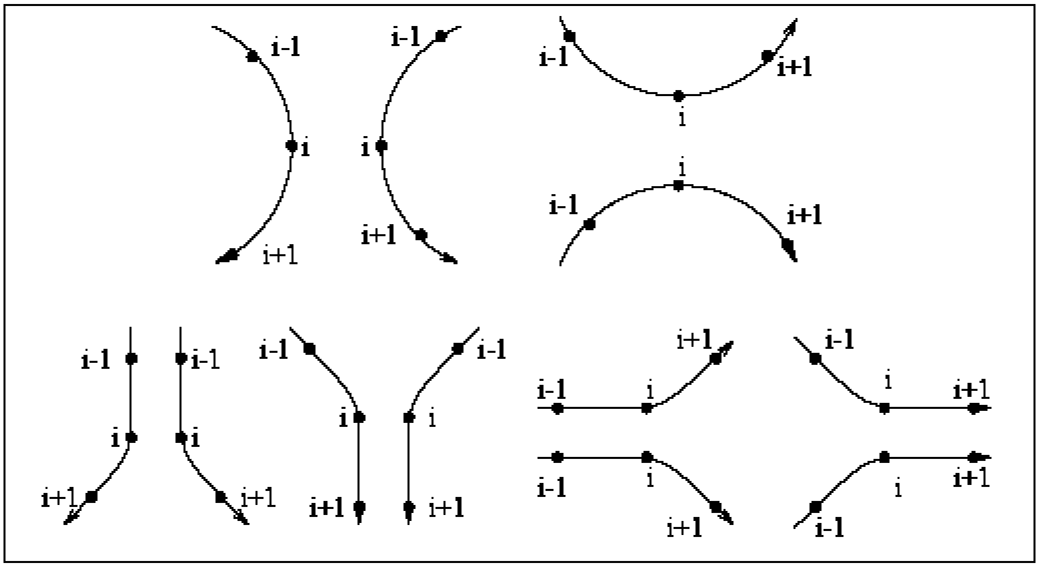


Figure 2.3: Critical Points [15]: Point i is the critical point while points $(i - 1)$ and $(i + 1)$ are the preceding and succeeding points, respectively.

Once the critical points are defined and the corresponding velocities are recorded, the signature is finely resampled. This is done to ensure that the signature is smoothed uniformly. If smoothing was performed before resampling, points of high writing speed will be smoothed more than points in low velocity segments. The next step is to perform the Gaussian filtering in order to eliminate noise. During smoothing, the points will be adjusted and thus will no longer be equidistant. Accordingly, resampling is again performed in a more coarse fashion. There exists a trade-off in this decision; either resample at a very small distance and pay the cost of high computational load during the matching process or resample at a larger distance and pay the cost of decreased accuracy. This is a parameter that needs to be tested in order to find a good balance between matching time and accuracy. Finally, the strokes are connected to form one long stroke. The reason for this will be evident during the matching process.

2.2.2 Feature Extraction

This algorithm fits into the category of functional approaches as described by Planondon et al. [34]. One global feature is used throughout the matching process and this is the number of strokes in the signature. This value is recorded before the strokes are concatenated together during preprocessing and stored as a global feature. The rest of the feature extraction process attempts to retrieve local information about the signature. The local information can be divided into two categories; spatial (shape) and temporal (speed) features.

The spatial features investigated by the authors include:

- distance between two consecutive points, δx and δy
- absolute y-coordinate, y
- sine and cosine of the angle with respect to the x-axis between two consecutive points, $\sin \alpha$ and $\cos \alpha$
- curvature, β

- gray values in a 9x9 pixel neighborhood

Figure 2.4 displays how each feature is computed for point p_i. The two points preceding p_i are p_{i-1} and p_{i-2} and the succeeding points are p_{i+1} and p_{i+2}. The δx and δy features are computed with respect to the subsequent point p_{i+1}. The absolute y-coordinate is just the y-coordinate of the resampled point. The angle α is the angle between the x-axis and the line through points p_i and p_{i+1}. This is used to compute the $\cos\alpha$ and $\sin\alpha$ features. The angle α is not used as a feature itself because of the nature of its value. An example given by the authors is that angles 1 and 359 are very similar but if the angle is used as a feature, they are very far apart using the Euclidean distance as a metric. Accordingly, the cosine and sine are used to better represent the direction of writing where both are needed to fully express the angle of writing. The curvature feature, β, is the angle between the lines $\overline{p_i p_{i-2}}$ and $\overline{p_i p_{i+2}}$. When this is computed, some points needed at the beginning and end of the signature may not be present and the closest existing points are used to calculate the two lines.

Figure 2.5 shows an example of how the 9x9 grid of gray values is calculated. This space is divided into nine 3x3 squares. For each square, a gray value is computed by summing the pixel values of that window. In this calculation, a pixel value is either 1 or 0. Accordingly, these 3x3 squares can have values ranging from zero through nine but in practice normally range from zero through four. Performing this calculation, nine feature values are computed for each point. Including all the other spatial features, each point can potentially have its shape described by fifteen feature values.

On-line signatures not only provide the ordering of the points but also the speed at which each point is written. This speed information is stored at each critical point and is then interpolated across the resampled points. In this algorithm, the is speed analyzed between the critical points and resampled points. Specifically, features are extracted for

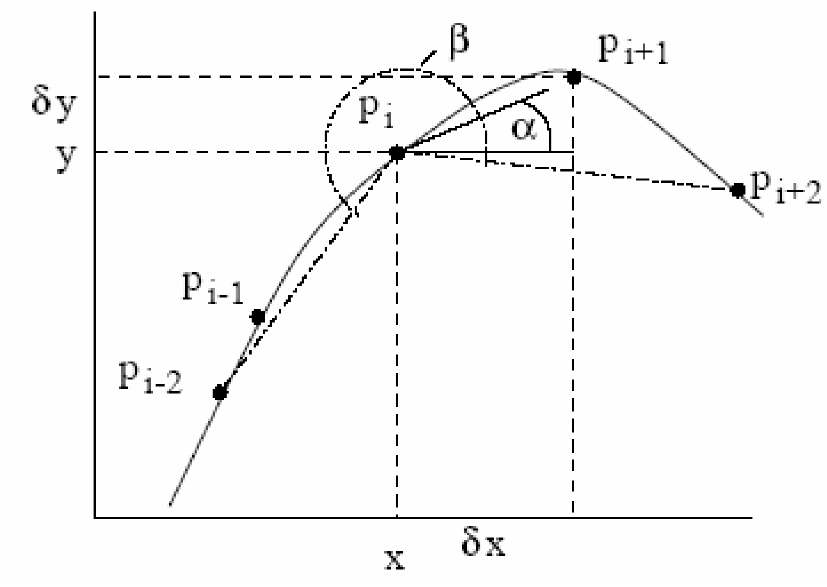


Figure 2.4: Features calculated for point p_i with respect to the shape of the signature [15].

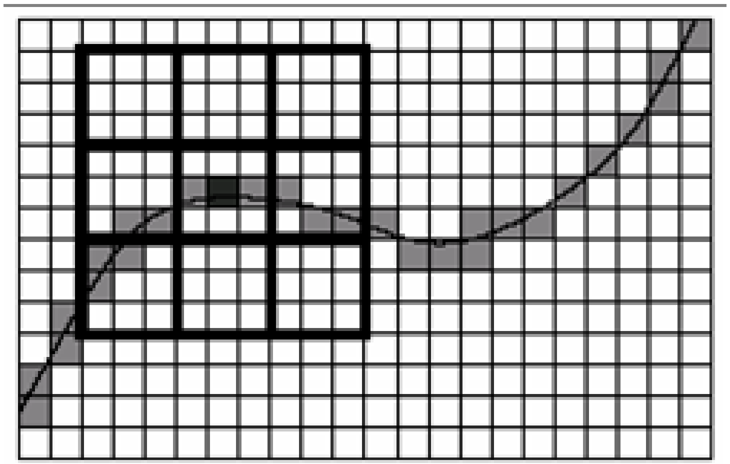


Figure 2.5: Gray values calculated in the 9x9 pixel grid [15].

- absolute and relative speed at each resampled point
- absolute and relative average speed between two critical points

The absolute speed for each resampled point is already calculated after interpolating the speed information between two critical points. The absolute average speed between two critical points also needs minimal computations, where the absolute speed is averaged from one critical point to the next. Relative speeds are investigated with the hope that although the absolute signing speeds may vary due to writing conditions, attitude, etc., the relative signing speed at the points should be more stable. The relative speeds are calculated by dividing the absolute speed at each point by the average writing speed over the whole signature.

2.2.3 Matching

After the local features are extracted from two signatures (a template and an input signature), these signatures need to be compared to find a difference or similarity measure between them. The algorithm used by the authors is dynamic time warping. This algorithm finds an alignment between two sets of points such that the sum of the differences between each pair of aligned points is minimal. To compute this difference value, the Euclidean distance is used as the metric. The results of this algorithm produce a set of pairings in the following fashion

$$\{(e^T_{t^T(1)}, e^I_{t^I(1)}), (e^T_{t^T(2)}, e^I_{t^I(2)}), ..., (e^T_{t^T(N)}, e^I_{t^I(N)})\}$$

where $e^T_{t^T(n)}$ is the $t^T(n)$th point in the template signature T, n is the position of the point in the alignment, and the function $t^T(n)$ returns the position of the point in the original string. Similar definitions apply for the points $e^I_{t^I(n)}$ of the the input signature I. The alignment of these pairings is done under the following constraints:

- $t^T(i) = t^I(j)$ if and only if $i = j$

- $t^T(1) < t^T(2) < ... < t^T(N_T)$

- $t^I(1) < t^I(2) < ... < t^I(N_I)$

The first constraint requires that there is a one to one relationship between matching pairs of points in the input and template signatures. The second and third constraints make sure that the alignment obeys the temporal ordering of the points in the signatures. For example, if point 1 in the input signature is matched with point 5 in the template, point 2 in the input signature cannot be matched with any of the points 1 through 5 in the template. Finally, to avoid solutions where a minimum distance of zero is found by not matching any of the points, spurious and missing point penalties are assigned. A spurious penalty is assigned if there is an extra point found in the input signature that does not match with any points in the template. Similarly, a missing penalty is assigned if there is no point in the input signature that corresponds with a point in the template. To find the alignment between two signatures, under these constraints, the minimum difference can be defined recursively:

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1, j-1) + d_E(i,j) \\ D(i-1, j) + MissingPenalty \\ D(i, j-1) + SpuriousPenalty \end{array} \right\}, \tag{2.4}$$

where $d_E(i,j)$ is the Euclidean distance between the feature vector for point i and the feature vector for point j. The final matching distance measure between a template T and an input I is then defined as

$$Dist(T, I) = D(N_T, N_I), \tag{2.5}$$

where N_T and N_I are the number of points in the template and input signatures, respectively. This score must be normalized to account for the number of points in each signature. Also, the final matching score should include the information from the global feature, namely, the number of strokes. Accordingly, the final distance score is

$$Dist(T,I) = \frac{Dist(T,I)^2}{Norm\_Factor(N_T, N_I)} + (SP)\left|S_T - S_I\right|, \tag{2.6}$$

where $Norm\_Factor(N_T, N_I)$ is the maximum possible distance between two strings of length N_T and N_I scaled by a constant factor, SP is the stroke penalty, and $|S_T - S_I|$ is the difference in the number of strokes between the template and input signature.

2.2.4 Enrollment and Verification

To enroll in the system, three template signatures are necessary. When an input signature is entered, it is compared against all three templates and either the minimum, maximum, or average of the three scores is used as the matching distance. Tests found that the minimum produced the best results. This score is compared against a common threshold and the input signature is classified as either genuine or impostor.

2.3 Improvements

In this thesis, various methods have been investigated in order to improve the performance of the signature verification algorithm in [21]. Enhancements to the preprocessing, feature selection, and matching modules have been made and some new modules, such as user-dependent normalization have been investigated. A design of the system is shown in Figure 2.6. Each of the additions/modifications to the algorithm in [21] will be explained in detail in the following sections.

2.3.1 Preprocessing

In order to distinguish between skilled forgeries and genuine signatures, a higher weight must be assigned to the temporal information in an on-line signature. This is because the shape of a signature can be easily reproduced by a forger but the temporal information

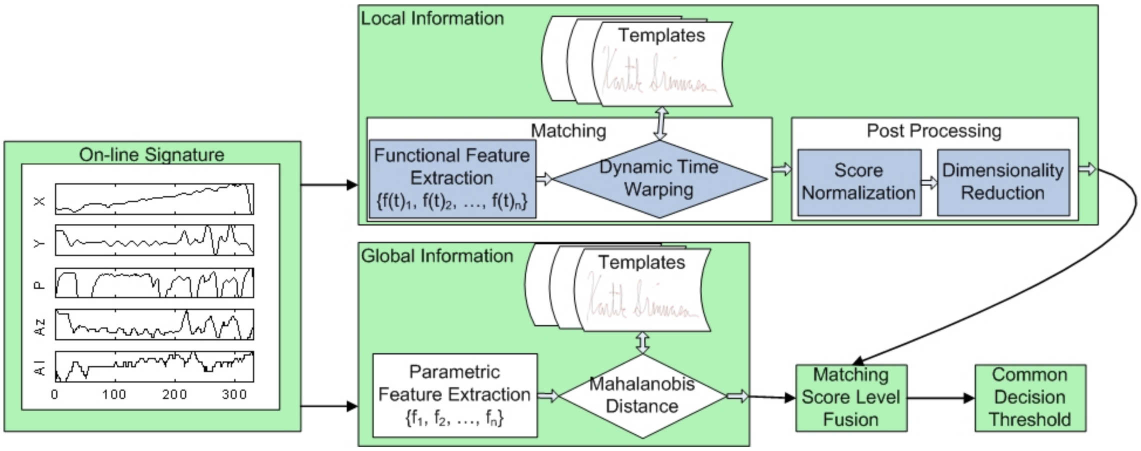


Figure 2.6: Block diagram of the signature verification system.

regarding timing and pressure is much harder to forge. When resampling is performed, the shape of signature can be more reliably compared at the expense of losing important timing information. The previous algorithm took the precaution of extracting critical points and preserving a subset of the temporal information at these points. This approach still loses much information when the speed is interpolated for the resampled points between critical points. It cannot be assumed that the speed is a linear function over time between two critical points. Accordingly, we no longer perform resampling. This seems like a more natural approach to match two signatures because handwriting is commonly modeled as ballistic motions. Using this type of model, one should concentrate on analyzing the forces in terms of strictly magnitude and duration [19]. This information can be better captured by analyzing the temporal features of a signature and concentrating less on the shape. When no resampling is done, all the temporal information is preserved and can be utilized in the matching process.

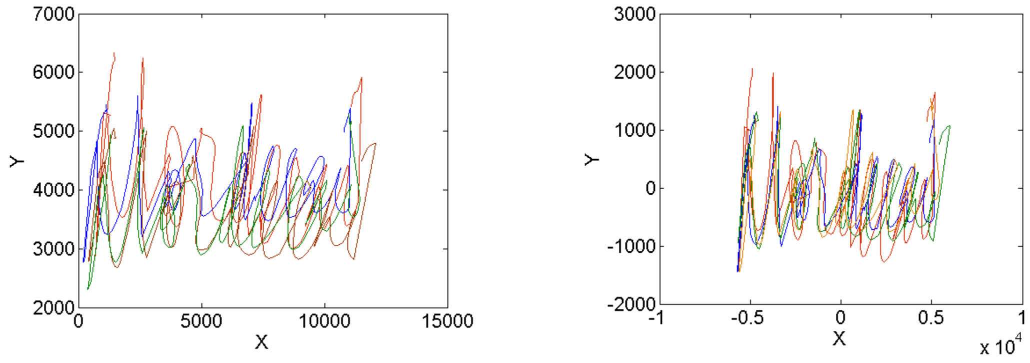
When no resampling is performed, this also effects the decision of whether the signature should be smoothed to eliminate noise. If smoothing is performed without resampling, this still preserves the temporal information but, as mentioned earlier, segments of high writing velocity will be smoothed more than segments written at a low velocity. Accordingly, smoothing is also no longer used during preprocessing.

A very simple, although useful, preprocessing procedure is to perform position (location) normalization. Some techniques normalize the position by transforming the signatures so that they have the same starting point. Another approach is to align the centers of the two signatures and this approach is used in our system. This is performed by subtracting the mean x and y coordinates of the signature from each individual point and is formalized in equations 2.7 and 2.8.

$$x' = x - \frac{\sum_{i=0}^{n} x_i}{n} \tag{2.7}$$

$$y' = y - \frac{\sum_{i=0}^{n} y_i}{n} \tag{2.8}$$

In the above equations, x and y are the original x and y coordinates, respectively, x' and y' are the transformed coordinates, and n is the number of points. The effect of position normalization is visually depicted in Figure 2.7.

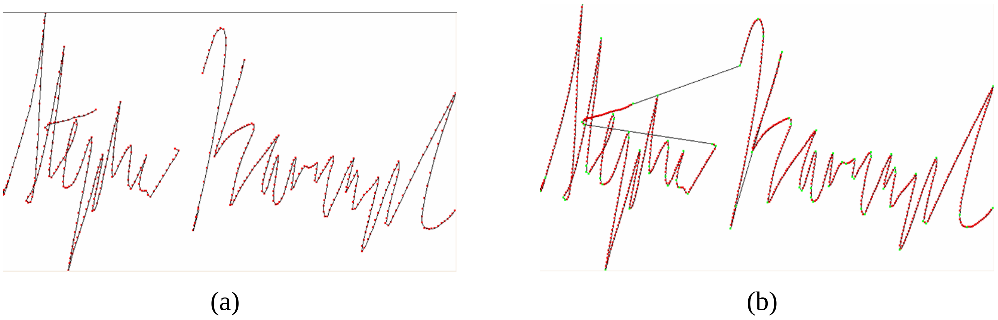


(a)

(b)

Figure 2.7: Position normalization; (a) signatures without position normalization, (b) signatures after position normalization.

Stroke concatenation is still used as in the original algorithm. This greatly decreases the overhead during matching. It is not necessary to find which strokes in an input signature correspond to which strokes in the template signature. If incorrect correspondences are found, this creates errors on top of any errors made while matching the points of each stroke. When one long stroke is created for both the input and template signatures, as shown in Figure 2.8, no correspondences need to be found at the stoke level and the focus can be placed upon matching the points.



(a)

(b)

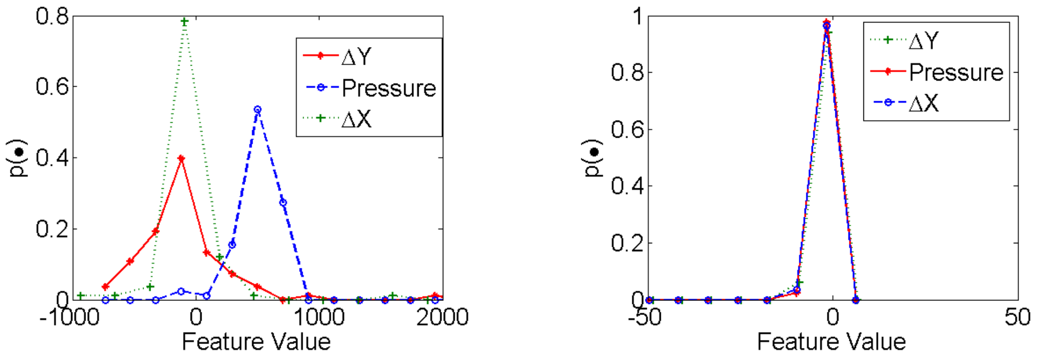
Figure 2.8: Stroke concatenation; signature before concatenation (a) and after the strokes are concatenated (b).

2.3.2 Feature Extraction

Feature selection was re-evaluated using the new preprocessing steps. Also, the pressure signal was considered as a feature. Experiments showed that the δx, δy and pressure features performed the best. In this case, the δx and δy features contain the velocity information, as opposed to shape information. Without resampling, the distance between two points in the signature is equivalent to the speed. This is because the signature is sampled at a constant rate and if this rate is used as the unit of time, the distance between points is equivalent to the velocity. Accordingly, δx is the horizontal velocity of the signature and

δy is the vertical velocity. The raw pressure signal is used without any transformation as a feature value.

A very important process during feature extraction is to normalize the feature values before attempting to match the feature vectors. Feature values can have very different distributions and if each feature is to be given equal weight, these distributions must be normalized. For example, the distributions of the $\cos \alpha$ feature is very different from that of the δx feature. $\cos \alpha$ ranges from -1 to 1 while δx normally ranges from -500 to 500. If both were provided to the matcher without normalization, δx would be given a higher weight because differences between this feature will result in a higher Euclidean distance than differences in $\cos \alpha$. A number of techniques have been proposed to perform normalization of this type such as min-max, decimal scaling, median absolute deviation, and tan-h estimators. In our work, we decided to utilize the most common normalization technique, called the z-score. The z-score is calculated using the arithmetic mean and standard deviation of the data, and is described in detail in Section 4.2.2. Here, we calculate the mean and standard deviation values according to each individual stroke. Because the mean and standard deviation are sensitive to outliers, computing the statistics within each stroke will limit the effect of the outlier strokes on the overall signature. This normalization technique does not guarantee a common range for the normalized scores but does ensure that the distributions of each feature will have a mean of zero and a standard deviation of 1. This effect is displayed in Figure 2.9. As can be noticed, this transformation only preserves the original distribution if it is Gaussian, due to the fact that the mean and standard deviation are the optimal location and scale parameters for Gaussians [22]. The assumption that features have Gaussian distributions is acceptable with our application.



(a) (b)

Figure 2.9: Feature normalization; (a) and (b) display the feature values before and after normalization, respectively.

2.3.3 User Dependent Normalization

As with other biometrics, especially behavorial biometrics, a small subset of the subjects may account for a large number of the observed errors. This phenomenon is normally due to these subjects' high intra-class variability (Figure 2.10). Some users may not be able to sign their name consistently or may sign their name in multiple fashions. When a common threshold on matching scores is used, subjects with high intra-class variability will often be falsely rejected because the input that they provide does not match closely with any of the signatures that they had enrolled with. Using a common threshold, both types of subjects, consistent and inconsistent, are penalized. This is because the inconsistent subjects drive up the threshold, making it easier than it should be to for a consistent user's signature to be forged. Also, the consistent users keep the threshold low, creating many false rejections for the inconsistent users. Accordingly some type of user dependent thresholding should be incorporated into the system to solve this problem. We use an approach proposed by Kholmatov and Yanikoglu [27]. To extract user dependent information, a sufficient amount of training data must be provided. In our system, 3-5 signatures are given to form the template for a user. When a user is enrolled, the pairwise distances between each of the template signatures is calculated using dynamic time warping, as will be described in the following section. After the pairwise distances of all the training signatures are computed, three normalizing statistics are stored:

- Average distance to the closest sample (I_{min})
- Average distance to the farthest sample (I_{max})
- Average distance to the template ($I_{template}$)

To compute the closest average distance, the distances between each sample and the sample closest to it are averaged over all the training data. To compute the farthest average distance, the same procedure is followed with the exception that the distance to the sample farthest from each training sample is averaged. The template is defined as the sample that has the minimum average distance from all other samples. This can be calculated by averaging the distances for each sample and selecting the sample with the lowest average as the template. This template sample is recorded along with its corresponding average distance. These statistics are store along with each training signature to create a profile for a user.

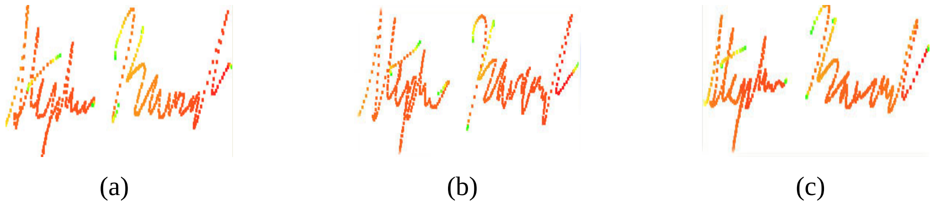

(a) (b) (c)

Figure 2.10: Signature intra-class variability. (a), (b), and (c) are three signatures from a single user.

When a is signature given as an input to the system along with a claimed identity, the input signature is matched with with each training signature stored in the profile for that identity. The minimum (P_{min}) and the maximum (P_{max}) distance scores from the input and training signatures comparisons are stored. Also, the distance score from the input to the preselected template is also recorded ($P_{template}$). These scores are normalized by their corresponding statistics that were previously discussed. This is calculated as follows:

$$N_{max} = I_{max}/P_{max} \tag{2.9}$$

$$N_{min} = I_{min}/P_{min} \tag{2.10}$$

$$N_{template} = I_{template}/P_{template}, \tag{2.11}$$

where N_{max}, N_{min}, and $N_{template}$ are the normalized scores of the maximum, minimum, and template distances, respectively. This results in a three dimensional vector of the form

$$\begin{bmatrix} N_{max} \\ N_{min} \\ N_{template} \end{bmatrix}. \tag{2.12}$$

A visualization of the comparisons made and statistics computed is shown in Figure 2.11. This normalization performs the same task as that of using user specific thresholds. User specific thresholds are an alternative to using a common threshold. Each user is assigned a threshold that depends on the variability of that user's signature. Users with high variability will be assigned a more lenient threshold while users with consistent signatures will be assigned a stricter threshold. This normalization works in the same fashion except that the scores are changed instead of the threshold. If a user has high variability in his signature, as evident from the training data, then the statistics calculated about the average distances will contain this information. The averages will be higher than that of a user with low variability. Accordingly, when an input signature is given that is not closely matched with any of the template signatures, resulting in high maximum, minimum, and template distances, these distances will be divided by the normalizing statistics that are also high. A similar situation occurs with a user with low variability, except that the resulting scores and statistics will now be lower. This normalization will then result in genuine score vectors clustering around the vector $[1\ \ 1\ \ 1]^T$. A common threshold can now be used when comparing the matching scores.

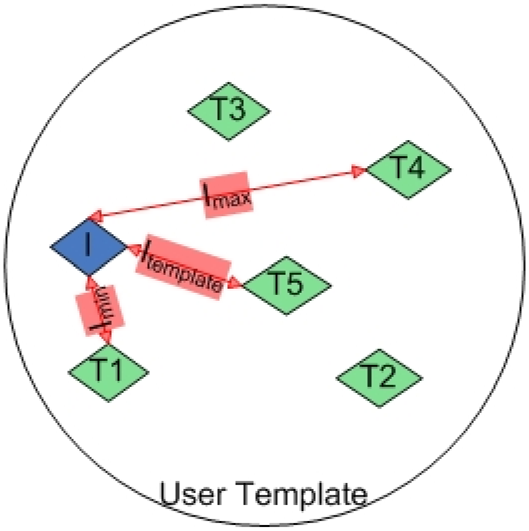


Figure 2.11: User-dependent normalization. I is the input signature and T_i, $i = \{1, ..., 5\}$ are the 5 stored templates for the claimed identity. The distances I_{min}, I_{max}, and I_{template} are used to construct the three-dimensional score vector. T_5 is selected as the template because it has the smallest average distance to all the other samples.

2.3.4 Dimensionality Reduction

The next step in the matching process is to transform the three-dimensional score vector down to one dimension. The first reason this is done is because the scores are highly correlated. If the maximum distance is high, then it is also likely that the minimum and template distances are also high. Accordingly, a large amount of the data's variance can be retained if a proper dimension reduction transformation is used. A second reason this is performed is so that a single, one-dimensional, threshold can be applied to the matching scores.

PCA

One of the most common dimension reduction techniques is Principle Component Analysis (PCA), also known as the Karhunen-Loeve Transformation. This algorithm attempts to find a linear transformation (W) that maps the original vector (X) to the projection vector (Y). Mathematically, this can be stated as

$$Y = W^T X. \tag{2.13}$$

The solution (W) consists of three steps. The first is to compute the average vector; u,

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{2.14}$$

where N is the number of data points, and x_i is the i^{th} feature vector. Using this average vector, the scatter matrix of the data can be calculated in the following fashion

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T. \tag{2.15}$$

This scatter matrix is simply the product of $N - 1$ times the sample covariance matrix. The direction of projection is an eigenvector of this scatter matrix. To extract the eigenvectors, the following equation must be solved to find the eigenvectors; e,

$$S_T e = \lambda e, \tag{2.16}$$

where λ is is an undetermined multiplier called the eigenvalue. The eigenvectors are ranked in order of their corresponding eigenvalues. Only a subset of the eigenvectors with the highest eigenvalues are used in dimensionality reduction. In our system, we selected the eigenvector with the highest eigenvalue as the direction of one-dimensional projection.

PCA requires training data in order to determine the projection vector(s). This presents the need of a validation set; an independent set of data that will not be used for either training the system (e.g. creation of templates) or testing the system. Consequently, our approach required the available data to be partitioned into three independent subsets: training, validation, and testing. The validation set must consist of both genuine signatures and skilled or random forgeries. This is because both genuine and impostor samples will be projected on the principal component axis during testing and in order to well separate the samples, both types of signatures must be provided during validation. As mentioned, genuine samples will cluster around the vector $[1\ 1\ 1]^T$ and will not produce a meaningful projection axis, where much of the variance will not be retained. If available, skilled forgeries are preferable over random forgeries to use while training PCA. If we can well separate genuine signatures from skilled forgeries along the principal axis, then it is very likely that the random forgeries will also be well separated from the genuine signatures. To calculate this projection vector, templates for each user must be created, as described in Section 2.3.3. Then, genuine and impostor scores must be calculated by comparing the remaining signatures of the validation set against the templates. These scores are given as the data to train PCA, in their original three-dimensional form. We then use the eigenvector with the highest eigenvalue to project the test samples down to one dimension. The results of this process are shown in Figure 2.12.

LDA

The PCA algorithm treats all the data (genuine and impostor) equally when performing dimension reduction, finding principal components that are useful for representing the data. A drawback of this technique is that these components may not be useful for discriminating between data in different classes (i.e. genuine and impostors). Exploiting class information can be helpful when performing dimension reduction. Linear discriminant analysis (LDA), also known as Fisher linear discriminant (FLD), uses the class information. PCA attempts to find projection directions that are useful for representation, while discriminant analysis attempts to discover projections directions that are useful for discrimination.

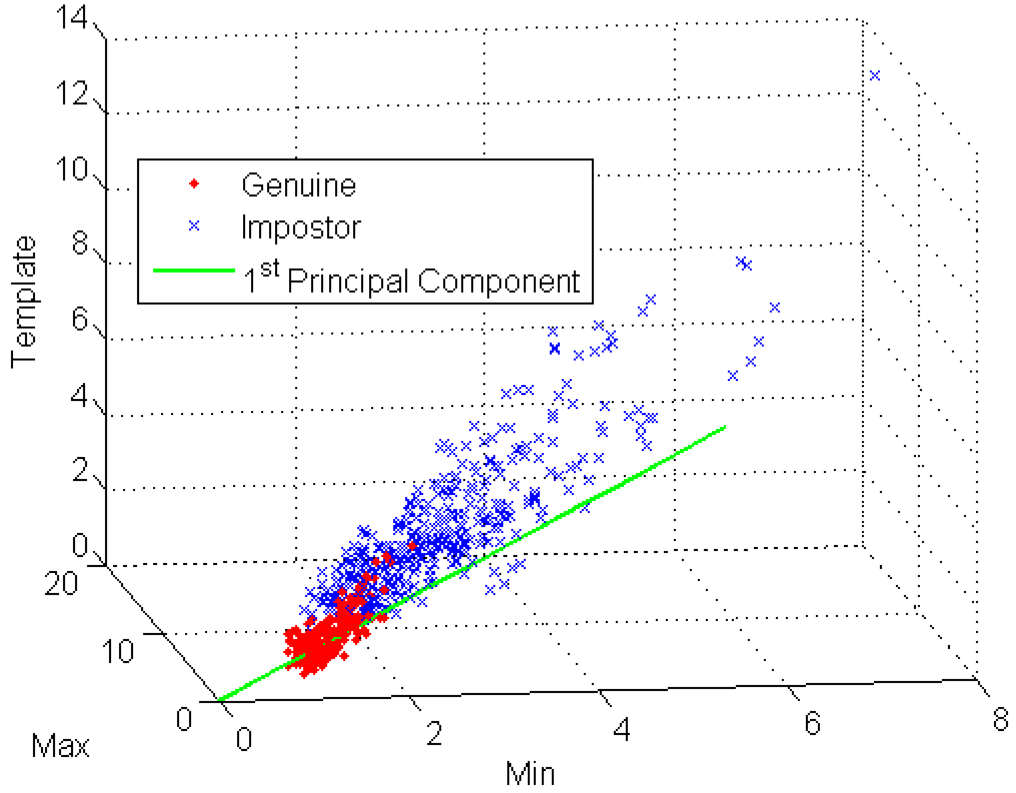


Figure 2.12: Principal Component Analysis: The genuine and impostor three-dimensional score values are plotted along with the first principal component calculated by PCA. The genuine scores are drawn as circles and impostor scores are as x's. These values were used as input to the PCA algorithm in order to produce the projection axis.

The LDA algorithm finds a transformation matrix W that maximizes the ratio of the between-class scatter matrix to the within-class scatter matrix. The between-class scatter matrix, S_B, and within-class scatter matrix, S_W, are defined as

$$S_B = \sum_{i=1}^{c} N_i(x_i - \mu)(x_i - \mu)^T \tag{2.17}$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T. \tag{2.18}$$

In the above equations, N_i is the number of training samples in class i, c is the number of distinct classes, μ_i is the mean vector of the samples belonging to class i, and x_i represents the set of samples belonging to class i [32]. Next, the transformation matrix W that maximizes the following ratio is estimated.

$$\frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \tag{2.19}$$

To calculate the optimal W that maximizes this ratio, eigenvectors and eigenvalues of the following equation are found

$$S_B w_i = \lambda_i S_W w_i, \tag{2.20}$$

where λ_i is the eigenvalues and w_i is the eigenvectors. The columns of the optimal W are the eigenvectors corresponding to the largest eigen values of the above equation [12].

In our application, we have only two classes, genuine, w_g, and impostor, w_i, so $c = 2$. The training data for dimension reduction is created in the same fashion as described for PCA except that class labels are also attached to each score vector. It is already known which samples are genuine and which are forgeries (random or skilled) in the validation set, so we are able to give all genuine scores a class label of 0 and all forgery scores a class label of 1. This data is then used as input to LDA to find a projection vector that best distinguishes between the two classes. Again, we used only the eigenvector that corresponds to the largest eigenvalue in order to reduce the dimensionality down to one. Results of this algorithm are shown in Figure 2.13.

In general PCA performed better than LDA in our experiments for two apparent reasons. First, because of the limited number of training samples that were available, LDA tended to over fit the training data and did not generalize enough to perform well on the testing data. The training data was not always linearly separable, in terms of the class labels, leading LDA to produce a projection vector that worked well for the training data but

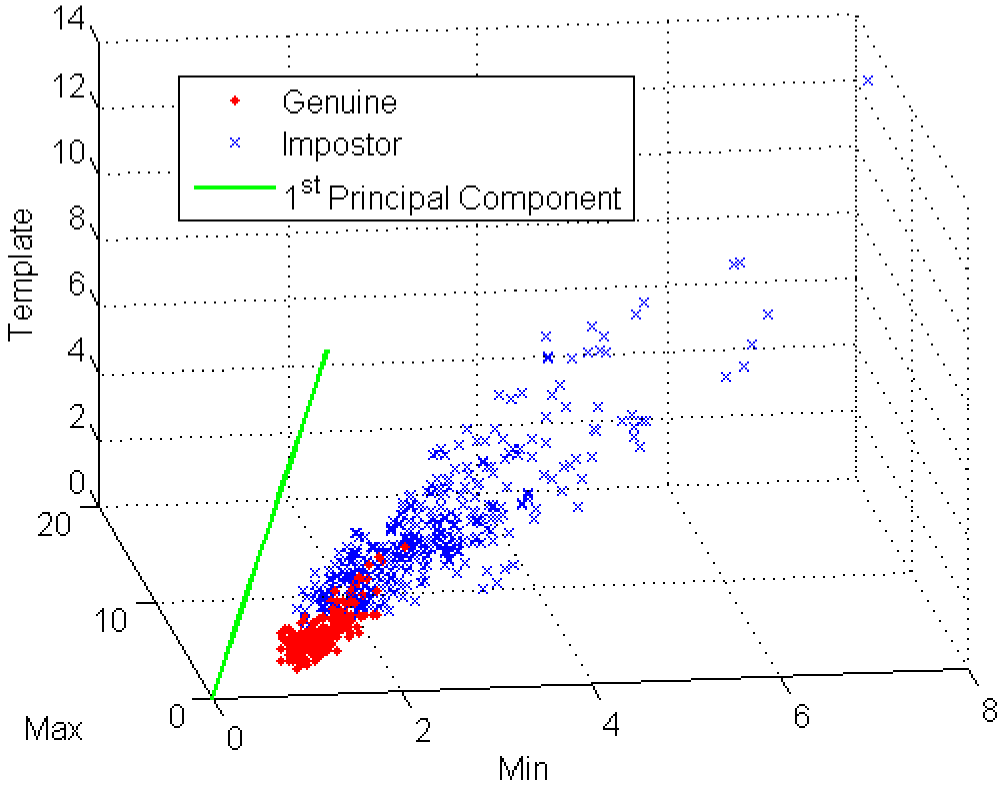


Figure 2.13: Linear Discriminant Analysis: The genuine and impostor three-dimensional score values are plotted along with the first principal component calculated by LDA. The genuine scores are drawn as circles and impostor scores are as x's. These values were used as input to the PCA algorithm in order to produce the projection axis.

not for the unseen signatures. The second reason is that the projection direction that best represented the data (by PCA) was also the same vector that best discriminated the data. As mentioned, the score vectors are highly correlated, where an increase of a feature value results in around the same order of increase in the other feature values. Genuine signatures will normally have values around 1 but will increase and decrease, depending on the consistency of the writer. Forgeries will generally have values larger than 1 and will span a wide range of values, depending on the accuracy of the forgery. If the projection axis followed the proposed pattern, then the two classes will be best separated on the axis that keeps the maximum variance among all the points. Accordingly, when PCA found the basis that best represented the data, it also found the most discriminating vector. Another important difference when using these two algorithms is that while PCA needed only skilled forgeries for training, LDA needs both skilled and random forgeries. Over fitting the data with just one type of forgery will lead to poor performance on the other.

2.3.5 Matching

The overall dynamic time warping algorithm remained, for the most part, unchanged from the algorithm in [21]. Only slight modifications were done to the parameters of the algorithm. To reiterate, the distance score for the dynamic time warping is calculated by solving the recursive formula

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1,j-1) + d_E(i,j) \\ D(i-1,j) + MissingPenalty \\ D(i,j-1) + SpuriousPenalty \end{array} \right\}. \tag{2.21}$$

The Missing Penalty and Spurious Penalty parameters are very important to the performance of the algorithm. Normally, these two parameters take on the same values because usually the cost of a missing point in the template or in the input signature is considered the same and can then be represented by a single parameter, called the gap cost. If the gap cost is selected too high, then the algorithm will be very rigid, resulting in low false accept rates but very high false rejects rates. If the gap cost is set too low, forgeries will be warped to closely fit the templates and many forged signatures will have low scores. This will result in low false reject rates but high false accept rates. Kholmatov suggests several strategies that can be followed in the selection of the gap cost [28]:

- Constant gap cost regardless of gap length
- Larger gap opening penalty followed by a much smaller gap extension penalty
- Gap cost increasing rapidly with gap length

- Different gap cost for reference and test signatures

The algorithm in [21] followed the third strategy. The gap cost increased linearly with the gap length. Here, we follow a different approach where the gap cost is more dynamic. It is calculated as follows:

$$GapCost = d_E(i, j) * c, \tag{2.22}$$

where $d_E(i, j)$ is the Euclidean distance between two feature vectors i and j, and c is a user defined parameter. The intuition behind this gap cost is that if the two points are very close, in terms of Euclidean distance, the cost of their mismatch (either a point is missing in the input signature or the template) should also be small. This allows the algorithm to find solutions, where if consecutive points are all very similar, that align these points in the best fashion and not enforce strict penalties if one of these points is missing. Also, if two points are very different, then an important part of the signature is missing and a high penalty should be enforced. The constant c, must be greater than one and its value should be estimated empirically. In our case, we set c equal to 1.5.

Another change that was made in the original algorithm was removing the missing stroke penalty in the final distance score calculation. The final distance score is now calculated by a more simpler equation:

$$Dist(T, I) = \frac{Dist(T, I)^2}{N_T N_I}. \tag{2.23}$$

The distance score is normalized by the number of points in both signatures, N_T and N_I. This was done because the stroke penalty will be incorporated into the global feature system described in the following section.

2.3.6 Global Feature System

The dynamic time warping algorithm takes the functional approach to signature verification and concentrates mainly on the local information. We took a parametric approach and combined this with the local information of the DTW algorithm. Global information can quickly increase performance by calculating simple features about the overall signature. For example, the duration of signatures is very discriminating between subjects. Although the calculations (local and global features) are being derived from the same data, we will show that the two approaches provide complementary information. Consequently, when the scores from local and global approaches are combined, the performance can be expected to increase.

Our global feature system calculates twenty features. These features are gathered from the list described in [14]. In [14], the authors calculated 100 global features and ranked them in terms of performance. We selected the top twenty of these features to use in our algorithm. The features that we use are displayed in Table 2.1.

Mahalanobis Distance

To match an input signature with a set of templates for a user, the Mahalanobis distance is used. This is a distance measure that utilizes the correlation between features. The Mahalanobis distance (d_M) between a sample x and a sample y is calculated using the following equation:

$$d_M(x, y)^2 = (x - y)' S^{-1} (x - y), \tag{2.24}$$

where S is the within-group covariance matrix. In this thesis, we assume a diagonal covariance matrix. This allows us to calculate the distance using only the mean and the variance

Table 2.1: Set of global features ordered by their individual discriminative ability. In the labels for the features, T denotes the time interval, t denotes the time instance, and N denotes a number of events. Other symbols are described in Table 2.2

Number	Feature Description	Number	Feature Description
1	signature total duration T_S	2	N(number of pen ups)
3	N(sign changes of $\frac{dx}{dt}$ and $\frac{dy}{dt}$)	4	average jerk j
5	standard deviation of a_y	6	standard deviation of v_y
7	(standard deviation of y)/Δy	8	N(local maxima in x)
9	standard deviation of a_x	10	standard deviation of v_x
11	j_{rms}	12	N(local maxima in y)
13	t(second pen down)/T_S	14	$\overline{v}/v_{x,\max}$
15	$\frac{A_{\min}}{\Delta_x \Delta_y}$	16	$(x_{\text{last pen up}} - x_{\max})/\Delta_x$
17	$(x_{\text{first pen down}} - x_{\min})/\Delta_x$	18	$(y_{\text{last pen up}} - y_{\min})/\Delta_y$
19	$(y_{\text{first pen down}} - y_{\min})/\Delta_y$	20	$(T_W \overline{v})/(y_{\max} - y_{\min})$

Table 2.2: Interpretations for symbols in Table 2.1

Symbol	Definition
jerk j	time derivative of the acceleration
a_y	acceleration in the y direction
a_x	acceleration in the x direction
v_y	velocity in the y direction
v_x	velocity in the x direction
Δ_y	$\sum_{i=1}^{\text{pen downs}} y_{\max\,\vert i} - y_{\min\,\vert i}$
Δ_x	$\sum_{i=1}^{\text{pen downs}} x_{\max\,\vert i} - x_{\min\,\vert i}$
$x_{\max\,\vert i}$	maximum x value in the i^{th} stroke
$x_{\min\,\vert i}$	minimum x value in the i^{th} stroke
$y_{\max\,\vert i}$	maximum y value in the i^{th} stroke
$y_{\min\,\vert i}$	minimum y value in the i^{th} stroke
j_{rms}	root mean square of the jerk
$\overline{v}$	average of the velocity
$A_{\min}$	$(y_{\max} - y_{\min})(x_{\max} - x_{\min})$
$y_{\max}$	maximum y value
$y_{\min}$	minimum y value
$x_{\max}$	maximum x value
$x_{\min}$	minimum x value
T_W	total time duration of all pen downs

of the features. Assuming a diagonal covariance matrix, the Mahalanobis distance calculation simplifies to

$$d_M(x, y) = \sqrt{\frac{(x - \mu_y)^2}{\sigma_y + k}}, \tag{2.25}$$

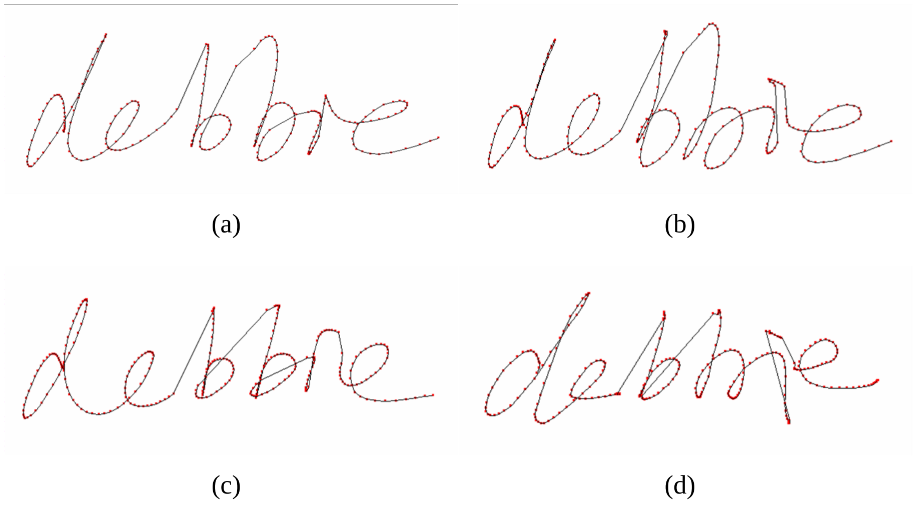
where μ_y is the mean vector of the population that y is sampled from, σ_y is the variance vector of the population, and k is a constant. The constant k is used to prevent the distance from becoming arbitrarily large when σ_y is very close to zero. This is a problem if too few training samples are provided to estimate the variance.

Because the Mahalanobis distance utilizes the feature covariance, no prior normalization for the features is necessary. This is a great advantage to using this as a distance metric. The mean and variance of each feature are taken into account during the calculation so that features with different ranges and distributions are all weighted equally.

2.3.7 Performance Evaluation

While a number of signature verification systems have been reported in the literature, they have never been compared against each other. Results are normally reported on a database that was gathered locally by the researchers. These databases differ in size and more importantly, difficulty. Some researchers are very diligent in gathering skilled forgeries to test the robustness of their algorithms while others test on only random forgeries. There was never any major effort to compare the different signature verification methods until the signature verification competition in 2004 (SVC2004) [43]. This competition created a benchmark signature database and protocols for conducting comparative studies. The competition itself used a database of 100 signers, each contributing 20 genuine signatures. 20 skilled forgeries were also created for each signer. The rules for testing verification systems make this a very difficult task, where a significant amount of the results are based on skilled forgeries. Currently, the data for the first 40 signers is publicly available and this will be the major database on which our algorithms will be tested on.

Tests are run on the SVC database using the following guidelines. Five out of the first 10 genuine signatures were randomly selected for training the system. Next, genuine scores were generated by testing on the following 10 genuine signatures. Skilled impostor scores were generated using all of the skilled forgeries for that signer (20 scores). Finally, 20 random signers are selected and a genuine signature from the random signer's set is used as a random forgery. This resulted in 10 genuine, 20 skilled, and 20 random scores for each signer. Ten trials of the above test were run, selecting different random signers and the average statistics were presented. In all of our tests, the same random signers are used for a fair comparison of the system. For tests with only 3 training samples, we used the first 3 out of the 5 random samples selected for training. For tests with 10 training samples, we used the first 10 genuine signature for training (no random selection was necessary). Some examples of the signatures in the database are presented in Figure 2.14.



(a)

(b)

(c)

(d)

Figure 2.14: Signatures in the SVC database; (a), (b) Genuine signatures, (c), (d) Skilled forgeries of the signatures shown in (a) and (b).

The performance of the original algorithm on the SVC database is shown in Figure 2.15. Ten trials were run using training set sizes of 3, 5, and 10. The performance of this algorithm is very poor on this database, especially on skilled forgeries. It should also be noted that as the size of the training set increases, the performance does not significantly increase.

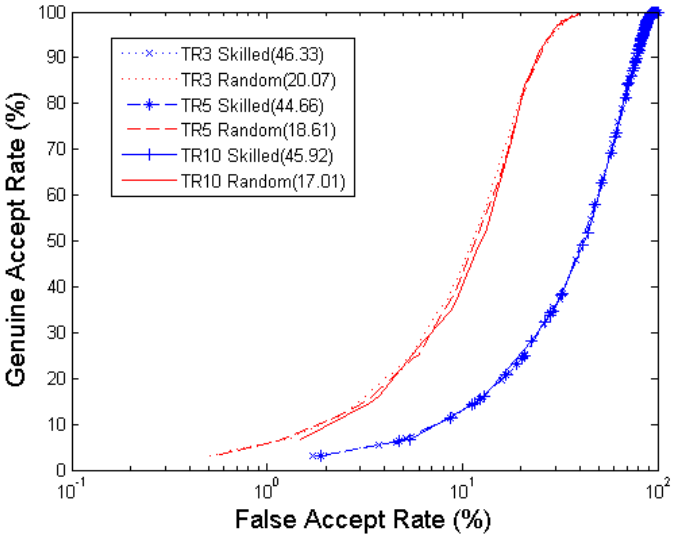


Figure 2.15: Average ROC Curves for the signature verification algorithm described in Section 2.2 on the SVC database. Tests were run on both skilled and random forgeries and the results are presented seperately. TR3, TR5, and TR10 are examples of testing with 3, 5, and 10 training samples, respectively. The equal error rate of each curve is shown in the parenthesis of the label.

The selection of the new features, along with the preprocessing stages, feature normalization, changes in the DTW, and user-dependent normalization greatly increase the performance. The best results are obtained using PCA dimensionality reduction. Results are shown in Figure 2.16. The performance of the algorithm is highly dependent on the amount of training data. When more training data is provided, the user normalization statistics are more closely estimated.

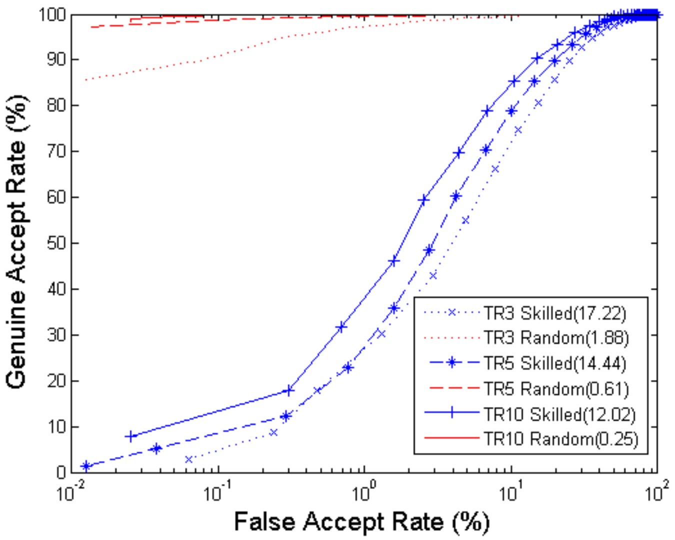


Figure 2.16: Average ROC Curves for the proposed signature verification algorithm on the SVC database using PCA dimensionality reduction. Tests were run on both skilled and random forgeries and the results are presented seperately. TR3, TR5, and TR10 are examples of testing with 3, 5, and 10 training samples, respectively. The equal error rate of each curve is shown in the parenthesis of the label.

The results for the global feature system are shown in Figure 2.17. The performance is not nearly as high as the local DTW approach but our main goal was to have this system provide complementary information about the signatures. Hopefully, the combination of the local and global systems will have better performance. A very noticeable factor that affects the performance is the size of the training set, even more so than the local system. This algorithm utilizes each of the training samples in calculating the final matching score and also to provide user-specific and feature-specific normalizations. When the size of the training set increases, the distribution of the feature vectors can be estimated more accurately.

The scores of the two systems are combined using the weighted sum rule, which will be described in detail in Section 4.3. The scores are normalized to a common domain using

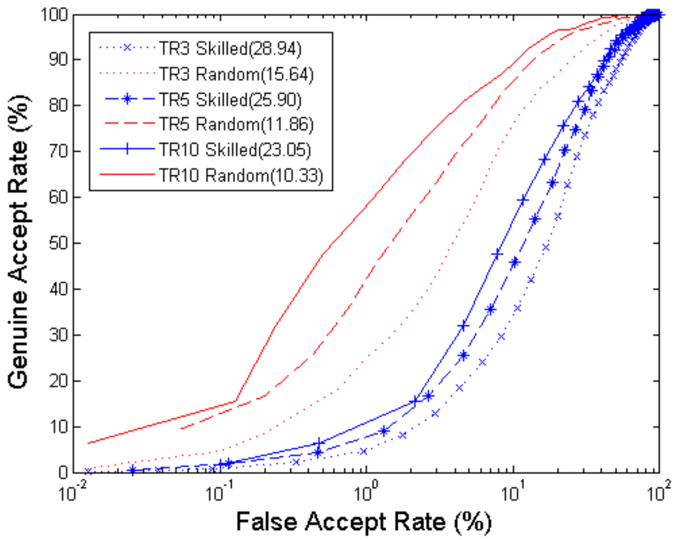
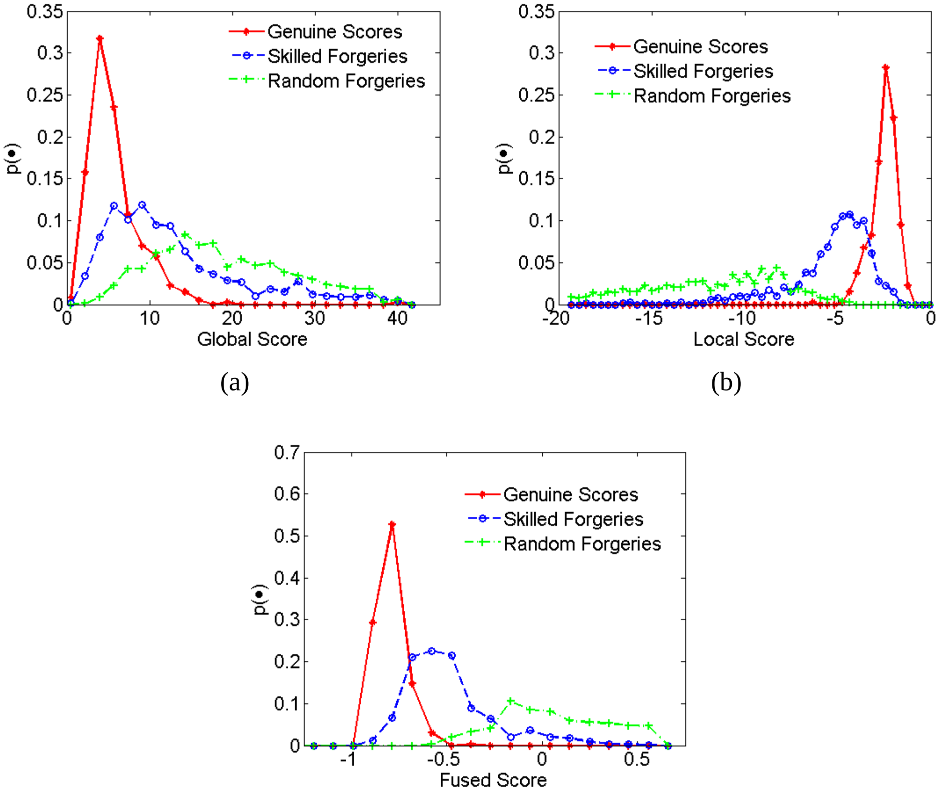


Figure 2.17: Average ROC Curves for the new signature verification algorithm using global features on the SVC database. Tests were run on both skilled and random forgeries and the results are presented seperately. TR3, TR5, and TR10 are examples of testing with 3, 5, and 10 training samples, respectively. The equal error rate of each curve is shown in the parenthesis of the label.

the z-score method, which is described in Section 4.2.2. The local scores are generated using the dynamic time warping algorithm with PCA dimension reduction. The global scores are generated from using the twenty global features and the Mahalanobis distance as the metric. The distributions of the global and local scores, before normalization are shown in Figures 2.18(a) and 2.18(b). A weight of 0.95 is assigned to the local score while a weight of 0.05 is given to the global score. The distribution of the fused scores is shown in Figure 2.18(c). The results of the combined systems are shown in Figure 2.19. These are generated using five training signatures. A performance increase occurs for the skilled forgeries, while the performance of the random forgeries remains the same.

In order to compare how this work relates to the state of the art, Table 2.3 displays the results published by the SVC competition in 2004. The signature verification algorithm



(a) (b)

(c)

Figure 2.18: Distribution of genuine and impostor scores from one trial of SVC testing; (a) Global (distance score), (b) Local (similarity score), (c) After sum rule fusion (distance score).

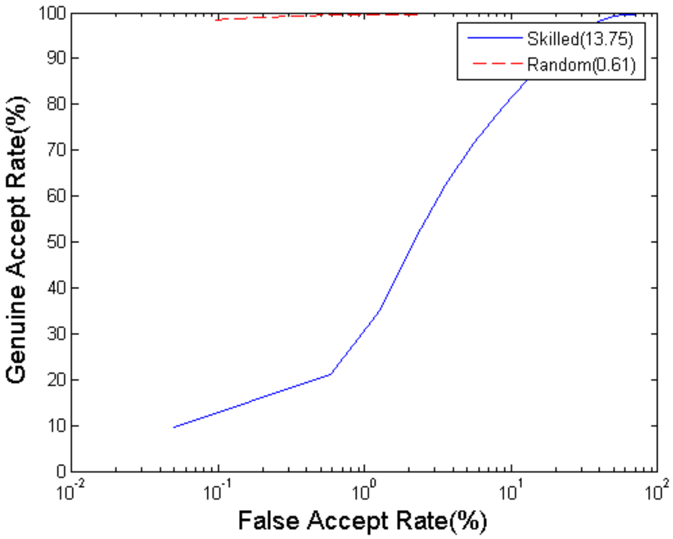


Figure 2.19: Average ROC Curves for the fused local and global scores on the SVC database. Tests were run on both skilled and random forgeries and the results are presented seperately. Five training signatures are used for testing. The scores are normalized using the z-score normalization and the weighted sum rule was applied with weight of 0.95 and 0.05 for the local and global systems, respectively. The equal error rates of each curve is shown in the parenthesis of the label.

developed in this work has an average EER of 13.75% on skilled forgeries and 0.61% on random forgeries.

Table 2.3: Results from the training data (first 40 users) from the SVC competition [43].

Number	Skilled EER	Random EER
1	6.90%	3.02%
2	6.91%	3.02%
3	6.96%	3.47%
4	7.64%	4.45%
5	8.90%	3.08%
6	11.29%	4.41%
7	15.36%	6.39%
8	19.00%	4.29%
9	20.01%	5.07%
10	21.89%	8.75%

CHAPTER 3

Speaker Identification and Verification

Our work utilizes both forms of voice recognition; verification and identification. This chapter will provide a brief overview of speaker identification and verification. Because we are using a commercial speaker verification system, we cannot provide exact details of how this specific system works, although much information can be derived from the documentation of the system.

3.1 Introduction

A generic speaker verification system is displayed in Figure 3.1. A microphone captures the sound from a speaker and this analog signal is converted to digital form. During enrollment the extracted features are used to create a template or a model for future verification sessions. During verification, the extracted features are compared with the model of a claimed identity to produce a matching score. Another dichotomy of speaker recognition systems is based on whether the recognition is text-dependent or text-independent. In text-dependent recognition, there is a required phrase that the person must say in order to be recognized. This phase is known by the system, being either fixed or prompted. A fixed phase may be something like a telephone or social security number, which changes very infrequently. Prompted phrases can be given either orally or visually. In either case, the verification system knows what words are being spoken as input to the system. In text independent recognition, the spoken phase can be anything that the speaker chooses to say.

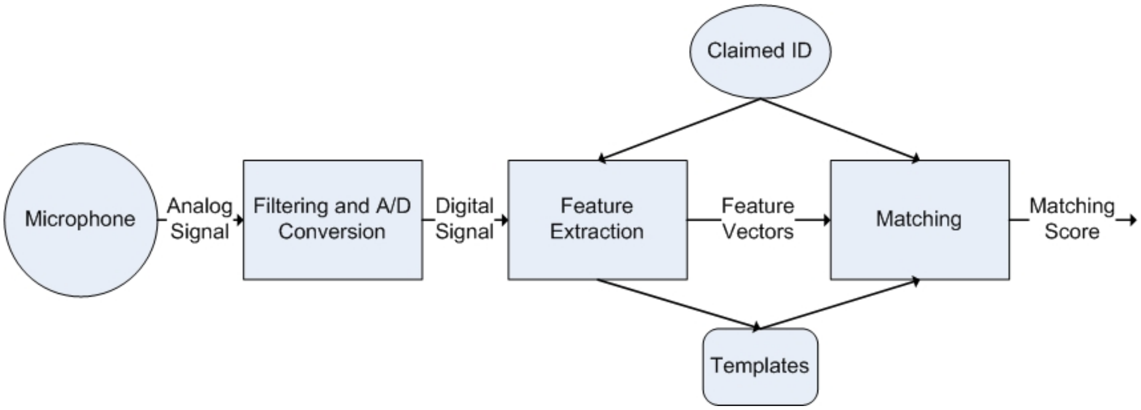
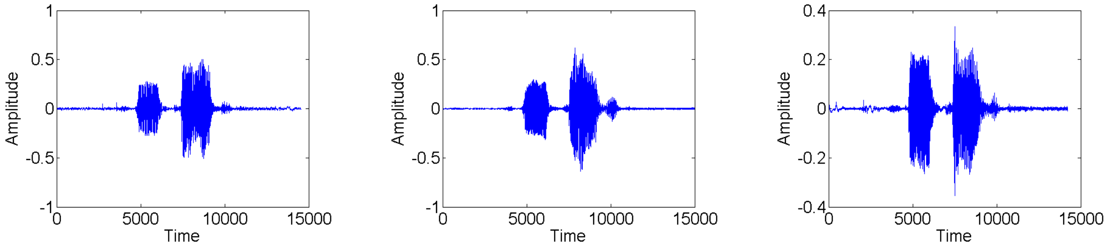


Figure 3.1: Design of a generic speaker verification system.

As with other biometrics, the performance of speaker recognition is very much dependent on the quality of the enrollment session. Before a user attempts to be recognized by the system, he must first enroll to create a voice model or template. The decision between text dependent or text independent modes must be made at this point. The more data that is presented to the system at the time of enrollment, the more complete the model will be for a user. When the data captured during enrollment is significantly different than the data presented for verification, it is reffered to as intra-class variability (Figure 3.2). For the voice biometric, the amount of data is measured by the total length of time of the voice samples given during enrollment. One must be careful of ambient noise and delayed versions of the voice entering the microphone from reflective acoustic surfaces during an enrollment session. Such factors are critical to the subsequent false accept/false rejection rates.

The major components of a typical speaker recognition system comprises of signal acquisition, preprocessing, feature extraction, matching, and decision making. Work on speaker recognition has been performed for many years and a variety of techniques have been explored. The most discriminating attribute among these techniques is in terms of



(a) (b) (c)

Figure 3.2: Voice intra-class variability. (a), (b), and (c) are three waveforms (amplitude vs. time) from a single user who spoke his first and last name three different times.

the matching procedure. The most well known techniques include artificial neural networks (ANN), dynamic time warping (DTW), hidden Markov models (HMM), and vector quantization (VQ). The Nuance system uses HMMs for its verification engine and will accordingly be the focus of the following sections.

3.2 Acquisition

Speech is an acoustic signal that is captured using either a microphone or a telephone. The signal is then anti aliased in order to limit the bandwidth. The result is sampled to create a digital signal by an analog to digital converter. The sampling rate for the database captured in this work is 16 bits of resolution (amplitude) at 8000 samples per second.

3.3 Preprocessing

In order to represent a speech signal in a more compact and less redundant form, speech parameterization is performed. To build a statistical model for use in HMM pattern matching, two popular techniques are filterbank-based analysis and the LPC-based method. These obtain a cepstral representation of speech, which is very useful for text-independent speaker verification.

3.3.1 Filterbank-Based Analysis

Filterbank-based analysis is comprised of performing pre-emphasis, windowing, performing the fast fourier transform (FFT), applying a filterbank and finally performing a cepstral transform to extract the cepstral vectors. This process is displayed in Figure 3.3. Pre-emphasis is done in order to enhance the high frequencies of the spectrum. It is performed by applying the filter

$$x_p(t) = x(t) - a * x(t-1), \tag{3.1}$$

where $x(t)$ is the digital signal of speech at time t, a is a constant less than 1, and $x_p(t)$ is the pre-emphasized signal at time t. This filter is not alway applied but can regain the high frequencies that were possibly lost during acquisition.

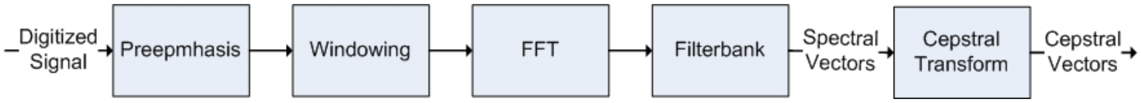


Figure 3.3: Sequence of processes performed during filterbank-based analysis.

Following pre-emphasis, the signal is windowed to extract local information about the signal. A window of fixed size is applied to the signal, starting at the beginning and moving up to the end, where the windows often overlap. Either a Hamming or Hanning window is used; these windows taper the original signal on the sides and consequently reduce the influence of the signals at the beginning and end of the speech sample. Each of these windows will provide spectral vectors to be used in the calculation of the cepstral coefficients.

The FFT of each individual window is calculated. The number of points for the calculation is usually a power of 2 that is greater than the the number of points in the window.

Next, the amplitude (modulus) of the FFT is taken over each of the points. Because the spectrum is symmetric, only the first half of the points are kept.

In order to eliminate some of the intra-class variability from the spectrum, a filterbank is applied. This filterbank is a series of bandpass frequency filters and is designed such that only the desired frequencies are kept. An obvious choice would be to keep only the frequencies that are audible to the human ear. The Bark/Mel scale defines this range and the corresponding frequency filters are defined by

$$f_{MEL} = 1000 * \frac{\log(1 + f_{LIN}/1000)}{\log 2}. \tag{3.2}$$

After the filterbank is applied to the spectrum, the spectral envelope is obtained. The log of the spectral envelope is taken. The final process is to transform the spectral vectors by the cosine discrete transform. This transform will yield the cepstral coefficients.

$$c_n = \sum_{k=1}^{K} S_k * \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], n = 1, 2, ...L. \tag{3.3}$$

In the above equation K is the number of log-spectral coefficients, S_k denotes the log-spectral coefficients, and L is the number of cepstral coefficients desired. The result of this process is a set of cepstral coefficients for each window.

3.3.2 LPC-based method

The other method for extracting the cepstral coefficients from the sampled voice signal is to use the LPC algorithm. The process consists of pre-emphasis, windowing the signal, the LPC algorithm and a cepstral transform to extract the cepstral vectors, as shown in Figure 3.4. This method models the voice signal by a linear combination of its past values and a scaled system parameter. A model that is often used is the auto regressive moving average (ARMA) model, which is simplified in an auto regressive (AR) model. The ARMA model can globally represent human speech production and then the speech signal can be described, in compact form, by the coefficients of the global model. This process is simplified by using an AR filter as opposed to the ARMA filter.

First, pre-emphasis is performed if desired. Then, the signal is windowed, as described in the filterbank-based approach. Next, the LPC algorithm is performed. As mentioned earlier, this approach models the speech signal as a linear combination of the past values:

$$s_n = -\sum_{k=1}^{p} a_k * s_{n-k} + G * u_n. \tag{3.4}$$

In the above equation, s_n is the current output, p is the prediction order, a_k are the model parameters (predictor coefficients), s_{n-k} are past outputs, G is a scaling factor, and u_n is the current input. This equation is simplified if the current output is approximated by only the past output samples:

$$\hat{s_n} = -\sum_{k=1}^{p} a_k * s_{n-k}. \tag{3.5}$$

When this simplification is made, some information is lost about the signal and this is defined by the prediction error, e_n,

$$e_n = s_n - \hat{s_n}. \tag{3.6}$$

The LPC algorithm finds the prediction coefficients (a_k) that minimize the prediction error in terms of the mean squared error. Details can be found in [11].

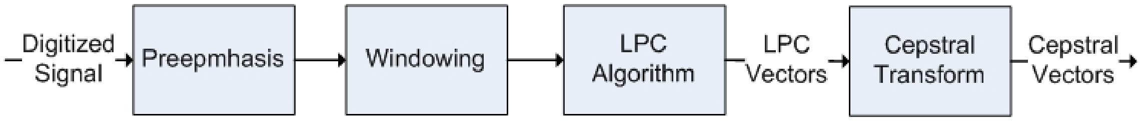


Figure 3.4: Sequence of processes that are used during LPC-based parameterization to extract the cepstral coefficients.

After the prediction coefficients are determined, the cepstral coefficients can be calculated by the following equations.

$$lc_0 = \ln \sigma^2 \tag{3.7}$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, 1 \leq m \leq p \tag{3.8}$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, p < m \tag{3.9}$$

In the above equations, σ^2 is the gain term, a_m are the LPC coefficients, and p is the number of LPC coefficients calculated.

3.4 Feature Extraction

After the LPC coefficients are extracted, they are manipulated in order to extract the useful information and discard any information that is not discriminative. One technique that is used is to center the cepstral coefficients. This is done by calculating the mean vector over all the windows and subtracting it from each vector. The variance of the vectors can also be normalized in a similar fashion.

Dynamic information about the vectors can be extracted by examining how they vary over time. The first and second derivatives of the signal are calculated by the following equations.

$$\Delta c_m = \frac{\sum_{k=-l}^{l} k * c_{m+k}}{\sum_{k=-l}^{l} |k|} \tag{3.10}$$

$$\Delta\Delta c_m = \frac{\sum_{k=-l}^{l} k^2 * c_{m+k}}{\sum_{k=-l}^{l} |k^2|} \tag{3.11}$$

Each of these time series can now be used as features corresponding to the voice.

The final step is to decide what information is important in distinguishing a speaker and what information should be discarded. One of the main goals of this step is to eliminate silence or background noise from the signal. This can be performed by computing a bimodal

Gaussian model of the feature vector distribution. The Gaussian with the lower mean is assumed to correspond to silence and background noise while the Gaussian with the higher mean is assumed to correspond to segments of speech. Accordingly, the portions of the signal that belong the the Gaussian with the lower mean are discarded.

3.5 Matching

A number of methods for matching have been used successfully in speaker recognition. The focus of preprocessing and feature extraction has been geared toward a statistical-based or stochastic-based model for use in HMM matching. This has recently been found to be very successful and is the basis of the Nuance recognition engine used in this work.

The difference between using a stochastic model versus a template model, as used with the signature verification system, is that the matching is probabilistic and the matching result is a measure of the likelihood of observing the given model. The problem then is to measure the likelihood of a collection of feature vectors given the speaker model. Two probabilities need to be computed for speaker verification; the probability that the feature vector is from the model of the claimed identity and the probability that the feature vector is not from the model of the claimed identity. The first probability is well defined and can be estimated directly from the training data of the claimed identity. The second probability is much harder to define because the entire space of all possible other identities must be modeled. A simple approach to this problem is to use the set of available speaker models to cover the space. The probabilities that the input speech matches each of these models is used with some function (e.g. average, min, max) to produce the probability that the speech does not match the claimed identity. A second and better approach is to train a single model from a pool of speakers. The main advantage of using this approach is that a single model can be trained and then used against all the claimed identities. The verification process using a background model for verification is shown in Figure 3.5.

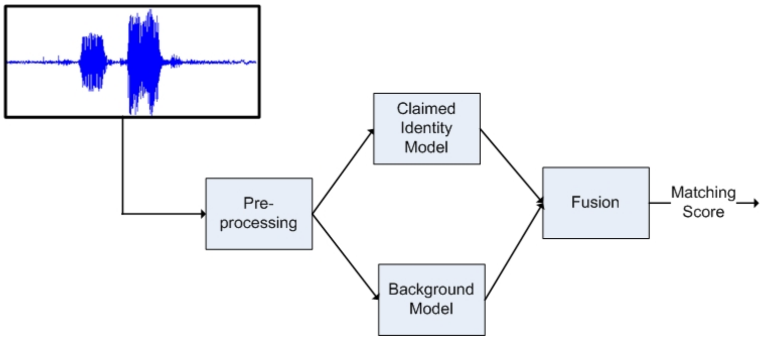


Figure 3.5: Speaker verification using a background model. A likelihood ratio is computed based on the probabilities generated from the claimed identity model and the background model to produce the matching score.

Gaussian mixture models have had the most success for text-independent speaker recognition. These model a D-dimensional feature vector x with a likelihood function that is a weighted sum of M Gaussian mixtures, given by the equation

$$p(x|\lambda) = \sum_{i=1}^{M} p_i b_i(x), \tag{3.12}$$

where $b_i(x)$ is the density for component i, and p_i is the weight of the i^{th} component. Each component density is a Gaussian defined by

$$b_i(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\}, \tag{3.13}$$

where μ_i is the mean vector and Σ_i is the covariance matrix. The mixture weights are constrained to add up to one. Overall, a GMM can be defined by the mixture weights (p_i), mean vectors (μ_i), and covariance matrices (Σ_i). Normally, a diagonal covariance is used as opposed to a full covariance matrix. This makes training the system simpler and has also been shown to outperform systems that use a full covariance matrix. A depiction of a Gaussian mixture density is shown in Figure 3.6.

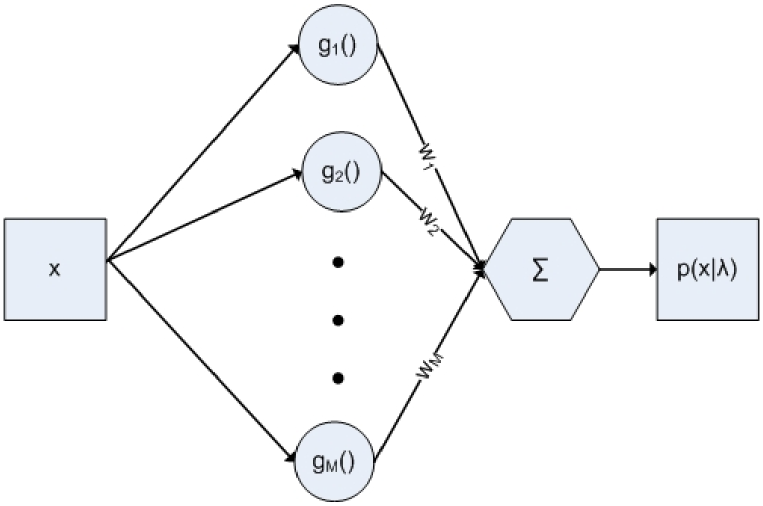


Figure 3.6: Diagram of a Gaussian mixture density representing the model for identity λ. g_i is the i^{th} Gaussian component, and w_i is the associated mixture weight. i ranges from 1 to m, where m is the number of components. Each Gaussian component is represented by a mean μ_i and covariance matrix Σ_i.

To train a model given training vectors derived from the speech signal of a user, the parameters of the GMM are manipulated in order to match the distribution of the training vectors. The most popular method for finding these parameters is maximum likelihood estimation. These parameter estimates can be obtained iteratively using the expectation-maximization (EM) algorithm. This algorithm increases the likelihood parameters of the estimated model in a monotonically increasing fashion until a threshold is reached. Details of the EM equations for evaluating these parameters can be found in [37].

After the models have been trained by estimating the parameters of the Gaussians, the probability of a sequence of feature vectors being produced by a claimed identity model can be calculated as

$$\log p(X|\lambda) = \frac{1}{T}\sum_{t} \log p(x_t|\lambda). \tag{3.14}$$

In the above equation, X is the sequence of feature vectors $\{x_1, \ldots, x_t\}$, λ is the model of the claimed identity, and T is the number of components. The average log-likelihood is computed for normalization purposes.

A technique that improves the performance of the GMM system is the use of the background model, or model that represents all other speakers. The background model is trained with speech from a variety of speakers. Individual speaker models are trained using the speech from that individual alone. An approach that increases the performance of the system is to adapt the parameters of the background model, using the speech of the individual attempting to enroll. The parameters can be manipulated using Bayesian adaptation or maximum a posteriori estimation. This provides the benefit of having the background model and the speaker's model be related in some fashion, rather than being completely independent. For details related to the adaptation of the parameters of the background model to create a speaker's model, refer to [10].

3.6 Score Normalization

As with signature verification, speaker recognition needs to take into account the intra-class variability (the variation of the voice of the same speaker) of the speakers. This variability can arise from differing quality of the enrollment data, the duration of the speech, sickness, emotional states, environmental noise, etc. To account for these factors, a single decision threshold cannot be used without prior normalization. Also, inter-class variability (the variation in the voices of different speakers) is an important factor when considering the placement of decision boundaries. Score normalization deals with the variability of the scores in order to make the placement of a decision threshold an easier problem.

A variety of normalization techniques have been proposed in the literature. Many of the ideas overlap with the normalization techniques proposed for the other biometrics, such as z-norm ($S' = \frac{S-\mu}{\sigma}$). Some techniques specific to speaker recognition have also be proposed, such as H-norm. In this approach, the attempt is to normalize for the type of handset that is used for telephone speech. Handset-dependent normalization parameters are estimated by testing genuine speaker models against handset-dependent speech signals produced by impostors [36]. When the test samples are input to the system, the corresponding normalization parameters for the specific handset are used for the score normalization. A number of other techniques have been proposed and it has been found that combinations of the normalization techniques produce better results.

3.7 Speaker Identification

Much of the presented material is in the context of speaker verification but it can also be directly applied to speaker identification. The difference is that instead of one claimed identity model λ, we are comparing an input utterance to a group of S speakers ($\{1, \ldots, S\}$) which are represented by S GMMs ($\{\lambda_1, \ldots, \lambda_S\}$). The goal is then to find the speaker model that produces the highest probability for the input feature vector sequence. Using logarithms and the assumption of independence between observations, the identification problem can be presented mathematically as

$$\hat{S} = \arg \max_{1 \leq k \leq S} \sum_{t=1}^{T} \log p(x_t | \lambda_k), \tag{3.15}$$

where $\log p(x_t|\lambda_k)$ is provided in Equation (3.14). This process is depicted in Figure 3.7.

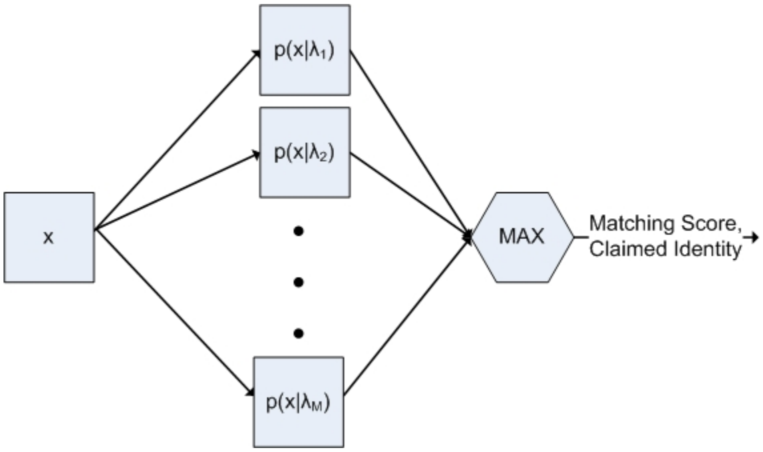


Figure 3.7: Speaker identification given an input feature vector x. λ_i is the mixture model for enrolled speaker i, where $1 \leq i \leq S$, and λ_m is the background model.

3.8 Nuance Speaker Recognition System

The Nuance speaker recognition system used in this work uses GMMs as the basis for its recognition engine. The system can work in a text-dependent, text-prompted, or text-independent mode. Our system works in the text-prompted mode where it is not initially known what the user will say during training. The system expects the same phrase as spoken during training to be repeated during verification.

3.9 Performance Evaluation

The performance of the Nuance speaker recognition system is reported in Chapter 4. The voice system obtains an equal error rate of 2.2% on a database of 100 individuals that each contributed 10 voice samples. Each of the samples was about 2 seconds in length.

CHAPTER 4

Multimodal Authentication

When using multiple biometric systems, combining information provided by the multiple biometric sources will normally lead to a higher recognition accuracy than a single biometric modality. In this thesis, we have used the voice and signature modalities together to improve the overall performance of the system and to make spoofing the system a more difficult task.

There are a variety of scenarios in which fusion of biometric modalities is necessary [38]. These include:

- A single biometric trait using multiple sensors. In this situation, multiple sensors record the same biometric trait. An example would be to have both an optical and capacitive sensor capture a fingerprint of the same finger or acquiring both 2D and 3D representations of the same face.

- A single biometric trait using multiple classifiers. Only one biometric trait is acquired and this is processed in multiple ways using different types of matchers. An example of this was described earlier where we acquire one signature from the signer but used both a local and global matcher on the same signature.

- A single biometric trait using multiple units. This scenario cannot occur with all biometrics. The same biometric trait is used along with the same sensor but there can be multiple inputs of the same trait to the sensor. This can be done with fingerprints by providing two separate fingerprints or in case of the iris by providing both eyes to the sensor.

- Finally, we can fuse multiple biometric traits. This is the most common scenario where more than one biometric trait is presented to the system, such as signature and voice. The system must combine the two modalities at some level in order to arrive at a single decision (accept or reject).

The information from multiple biometric systems can be integrated at three main levels; feature, matching, or decision level. Although, the available information can be fully utilized if fusion is performed at the feature level, this approach has several problems. First, there many not be any clear relationship between the feature spaces of the individual biometric systems. This could lead to the use of highly correlated features. A second problem is the “curse of dimensionality” [12]. A common approach is to concatenate the feature vectors of the different systems together to create a single feature vector. This may lead to a feature vector of long length, requiring a very large number of training samples to provide good performance. Finally, when using a commercial biometric product, access to the feature vector is restricted, so the fusion must be performed at a later stage in the processing.

A higher level at which the information of can be combined is at the matching score level. This level utilizes the most information about the individual systems, next to feature level fusion. Typically, biometric systems output a matching score and this is the level of fusion used in our system and will be the focus of this chapter.

4.1 Database

The data used in our experiments is a truly multimodal database gathered from 100 individuals in our laboratory. Ten signatures along with ten voice samples were gathered for each subject during a single session. The individuals were asked to sign their name and speak their first and last names. The data was collected using a Toshiba Protege tablet PC;

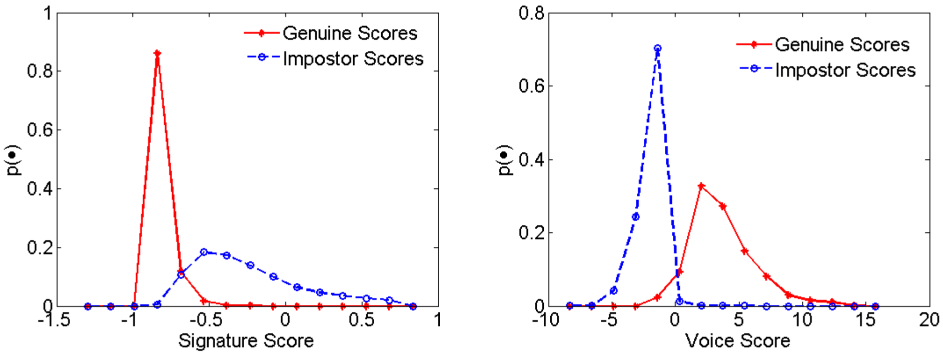
the stylus was used to gather the signatures and the internal microphone recorded the voice. The collection was done on our campus in various laboratories with significant background noise. The on-line signature data contains the x and y coordinates along with the pressure, altitude, and azimuth of the pen sampled at 100 points per second. The recorded speech was sampled at a rate of 8000Hz from one channel using 16 bits per sample.

Tests run on this database generated genuine and impostor scores for each user in the following fashion. Five random samples of voice and signature combinations were selected as the training set. Genuine scores from the remaining 5 samples, were computed. Impostor scores were generated by testing on one random sample from each of the 99 other users. This resulted in 5 genuine scores and 99 impostor scores per speaker. Cross validation was performed using this process 10 times (each time picking a different training set of size 5 for each speaker). Results are reported as the averages from these ten trials.

The distributions of the signature and voice scores are shown in Figure 4.1. The local and global systems for the signature matcher were combined in the same fashion as described in Chapter 2. The distributions and resulting ROC curves of this process are given in Appendix A.

4.2 Score Normalization

There are two common approaches to combine the matching scores of different biometric modalities; either classification or combination. In the classification approach, the scores of the individual systems are concatenated to form a feature vector. This vector is used as input to a classifier which will classify the feature vector into two classes: accept or reject. A benefit of using this approach is that the matching scores can be nonhomogeneous; one may be a distance measure while another can be a similarity measure and they may have different ranges. Consequently, no score normalization is necessary. A



(a) (b)

Figure 4.1: Distribution of genuine and impostor signature scores; (a) Signature (distance score) and (b) Voice (similarity score). Signature scores are based on the combination of local and global features.

drawback to using this approach is that additional training data is needed in order to train the parameters of the classifier in order to find a proper decision boundary.

The second approach is to combine the matching scores of the individual systems to generate a single score on which a decision threshold is set. This approach generally provides better performance but it must be ensured that the scores are properly transformed to a common domain before the combination. Score normalization attempts to solve the problems of non-homogeneous scores, varying score ranges, and differing distributions. It changes the location and scale parameters of the matching score distributions of different modalities so that all the scores share a common domain.

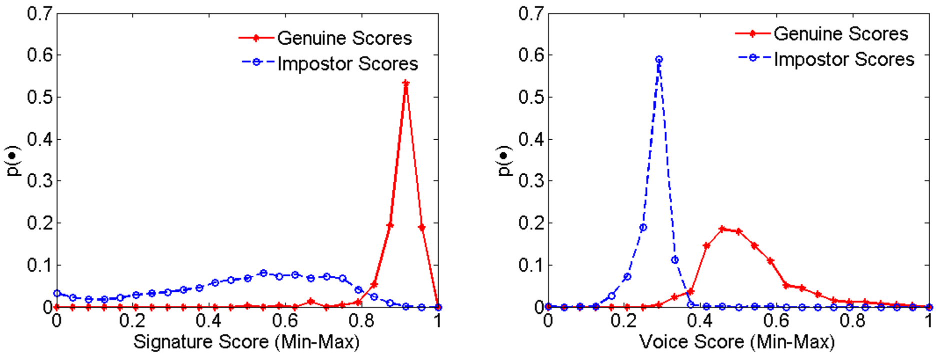
Various techniques have been proposed in the literature to normalize the matching score of biometric systems and it has been found that the min-max and z-score normalization techniques followed by a simple sum rule, generally outperform other techniques [22]. Accordingly, in this thesis, we experimented with the use of the min-max and z-score normalizations.

4.2.1 Min-Max

Min-max normalization is the simplest of the score normalization techniques. The normalization shifts the minimum and maximum scores to range between 0 and 1, respectively. If the minimum and maximum values of the matching score distribution are not known beforehand, they can be estimated given a set of matching scores. Formally the min-max rule can be defined as

$$s' = \frac{s - \min}{\max - \min}, \tag{4.1}$$

where s is the raw matching score, and s' is the normalized matching score. To account for non-homogeneous scores, a distance score can be transformed into a similarity score by subtracting the normalized score from 1 (1-s'_k). This normalization does not change the underlying distribution of the data except for a scaling factor. If the minimum and maximum values have to be estimated, outliers will effect the normalization. Figure 4.2 shows the results of transforming the signature and voice scores using the min-max normalization.



(a)

(b)

Figure 4.2: Distribution of genuine and impostor signature scores after min-max normalization; (a) Signature and (b) Voice.

4.2.2 Z-Score

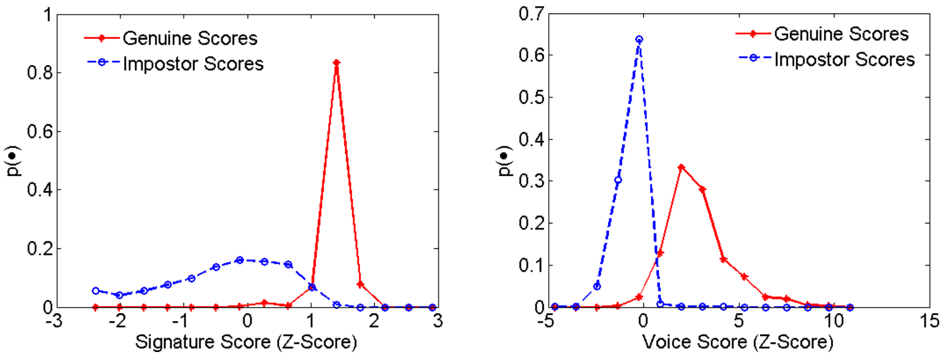
The z-score normalization technique is the most commonly used. It utilizes the mean and standard deviation of the data to normalize the scores. A normalized score is produced by the equation

$$s' = \frac{s - \mu}{\sigma}, \tag{4.2}$$

where μ is the mean and σ is the standard deviation of the matching score distribution. Similar to min-max, this method is also sensitive to outliers. This is because outliers can greatly affect the calculation of the mean and standard deviation and thus altering the transformation of the normalized scores. It differs from the min-max method in that it does not guarantee a set numerical range. This method attempts to change the score distributions so that they have a mean of zero and a standard deviation of one. If the distributions are not originally Gaussian, the transformation will not retain the original distribution. Figure 4.3 shows the results of transforming the signature and voice scores using the z-score. The signature score is converted into a similarity score using the transformation $x'_s = e^{-x_s}$, where x_s is the original signature score and x'_s is the converted similarity score. As can be seen, the distributions do not share a common range and the original shape of the score distributions of the data are not retained.

4.3 Score Fusion

Now that the matching scores share a common domain, they can be combined in a useful fashion. The problem of combining the scores from the voice and signature modalities for a given test sample T can be considered as a two class classification problem. The sample T can fall into either the impostor (w_i) or genuine (w_g) class. A Bayesian approach



(a) (b)

Figure 4.3: Distribution of genuine and impostor signature scores after z-score normalization; (a) Signature and (b) Voice.

would decide w_i if

$$P(w_i|x_v, x_s) > P(w_g|x_v, x_s) \tag{4.3}$$

and w_g otherwise. In the equation, x_v and x_s are the voice and signature scores, respectively, and $P(w|x_v, x_s)$ denotes the posteriori probability of a class given the voice and signature scores. The strategy used in our system is the simple sum rule described by Jain and Ross [23]. This rule assumes statistical independence of each modality and also that the posteriori probabilities computed by the individual classifiers do not deviate much from the prior probabilities [22]. The weighted sum rule assigns a test sample to w_i if

$$W_vP(w_i|x_v) + W_sP(w_i|x_s) > W_vP(w_g|x_v) + W_sP(w_g|x_s), \tag{4.4}$$

and w_g otherwise. In the equation, W_v and W_s are the weights of the voice and signature scores, respectively, and $P(w|x)$ is the posteriori probability of a class given a matching score.

4.3.1 Modes of Operation

Multimodal systems can perform fusion in one of three different modes; serial, parallel, or hierarchical mode [39]. In serial mode, the output of one modality is used to narrow down the search for the identities. This allows for the users to not have to simultaneously input all biometric traits at once and, also, this can lead the system to a decision before all modalities are input. In parallel mode, all the traits are input at the same time and the multimodal system uses all the scores together to make a decision. In hierarchical mode, the traits are combined in a treelike fashion where results of a subset of the classifiers are combined before adding the information from the other classifiers. This mode will be most useful when a large number of biometric traits are present.

Our multimodal system works in serial mode. First, a spoken voice is input to the system. The speaker recognition system first performs speaker identification on the speech signal. If this input does not match closely with any of the enrolled individuals, the user is classified as impostor and the on-line signature of the user is not obtained. If speaker identification finds a match to one of the enrolled speakers, this identity is passed to the signature verification system. The signature of the user is obtained, and this is compared against the identity found by speaker identification. When both the signature and voice scores have been obtained, score fusion will be performed.

4.4 Results

The weights (W_v and W_s) of the individual modalities have to be estimated from additional data (validation set). Our system uses common weights (for all the users) but using a user-specific specific weighting scheme may further increase the performance of the multimodal system [20]. Figure 4.4 shows the distributions of the signature and voice scores fused together using the weighted sum rule with min-max and z-score normalization. Figures 4.5 and 4.6 show the resulting ROC curves using min-max and z-score normalizations, respectively. The system using the z-score normalization gives better performance with an equal error rate of 0.72%. The weights used for the sum rule were 0.65 for the signature system an 0.35 for the voice system. These weights were determined from the validation set.

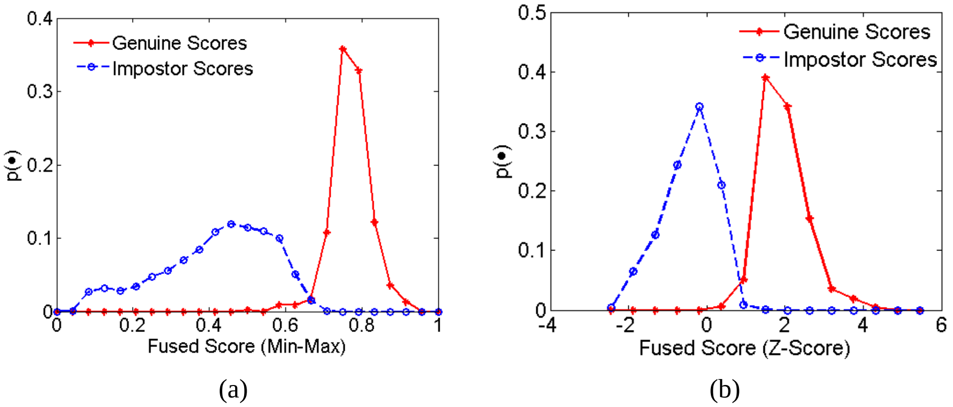


Figure 4.4: Distribution of genuine and impostor signature scores after sum rule fusion; (a) Min-Max normalization and (b) Z-Score normalization. The weights assigned to the signature and voice systems were 0.65 and 0.35, respectively.

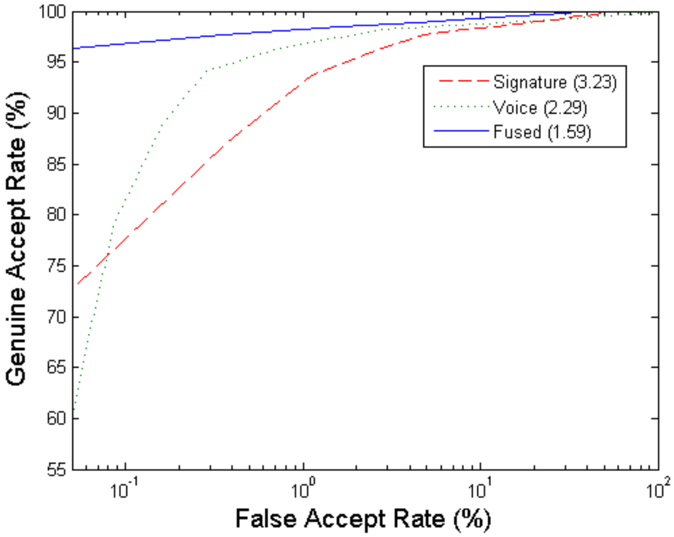


Figure 4.5: Receiver operating characteristic curves for the signature, voice, and fused system scores using min-max normalization. The equal error rate of each system is displayed in parenthesis in the legend.

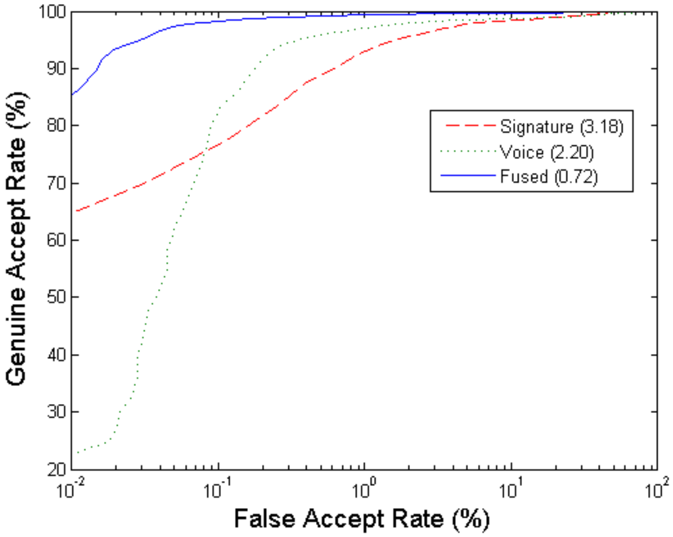


Figure 4.6: Receiver operating characteristic curves for the signature, voice, and fused system scores using z-score normalization. The equal error rate of each system is displayed in parenthesis in the legend.

CHAPTER 5

Summary

The motivation of this thesis is to provide a secure form of authentication for access to medical records. The use of biometrics provides increased security over standard forms of authentication because it relies on who we are and what we do. This type of information provides enhanced security levels because different individuals are unlikley to have similar physiological and behavioral attributes.

The use of the signature and voice modalities were selected for two main reasons. First, these are well accepted in the medical domain and emulate the current system of authentication in hospitals. Second, a tablet PC does not require any additional hardware in order to accommodate biometric authentication using signature and speech. Use of multiple biometrics overcomes some of the limitations imposed by unimodal systems and increases the performance of the authentication.

Signature verification was explored in Chapter 2. This utilized dynamic time warping to match feature vectors from two signatures. The system focused on the temporal information captured by the on-line signature to increase the performance in classifying skilled forgeries. We incorporated a form of user-dependent normalization to account for the intra-class variability presented by the signers. This required the use of dimensionality reduction techniques, with the best performance obtained using Principle Component Analysis, to transform the normalized score vector to one dimension. A global feature system was also added to provide complementary information about the signature. Twenty features were extracted that examined the signature in its entirety and the Mahalanobis distance was used to compute the distance between a user template and an input feature vector. The two scores (global and local) were combined using a simple weighted sum rule to provide a robust signature verification system.

Voice identification and verification was discussed in Chapter 3. To perform identification and verification, we used the Nuance Verifier system. This system relied on Gaussian mixture models to analyze the acoustic information provided in the voice signal. We worked in a text-prompted mode, where the speaker could say anything during training but had to use the same phrase spoken in training for subsequent verification sessions. This system was used to perform both identification and verification. A user attempting to access the system must first speak his first and last name. The voice system performed identification to recognize if the input speech matches any of the enrolled models. If multiple matches were found, verification was performed on each of the possible identities and the model with the highest similarity score was used as the claimed identity. This score was passed onto the fusion module while the claimed identity was passed to the signature verification system.

Chapter 4 focused on the techniques used to combine the matching scores of the two unimodal systems. First, score normalization was performed, experimenting with the min-max and z-score techniques. The normalized scores were then combined using a weighted sum rule to output a final matching score. This was compared against a common threshold to classify the input as either genuine or impostor.

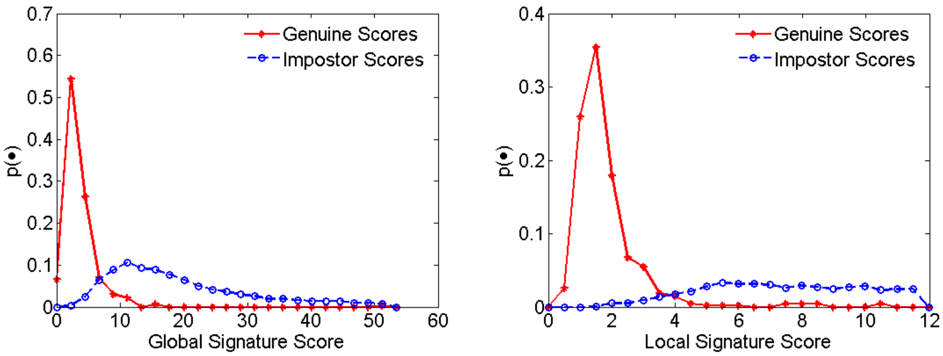
5.1 Future Work

The techniques presented in this thesis provide a robust multimodal authentication system to protect the privacy of medical records. Further work can be done to improve the performance of the system. Other forms of dynamic information from the on-line signature can be incorporated into the system. These can be in the form of calculating acceleration signals in the x and y directions and also the time derivatives of the acceleration signals. Also, experiments with the tilt of the pen (altitude and azimuth) can be explored to see if these provide distinguishing information for skilled forgeries. It is also important to further explore the area of user-dependent normalization techniques to allow the system to handle variations in the handwriting.

Further score normalization techniques should also be explored. The min-max and z-score normalizations were not robust to outliers in the data. Techniques that are not greatly effected by outliers may provide for better fusion results, such as tan-h normalization or Parzen window density estimation. The fusion systems can also be further explored. Classifications techniques such as support vector machines, may be able to find a better decision boundary than using a simple sum rule. It can also be beneficial to use user specific weights when using combination fusion. It may be the case that while one user has very reliable signature scores (allowing higher weight to be put upon the signature system's score) other users may have very high intra-class variability for their signature and the multimodal system may obtain better performance if most of the weight was placed upon the voice modality. Also, the system should be tested on a larger database to validate the robustness of the algorithms.

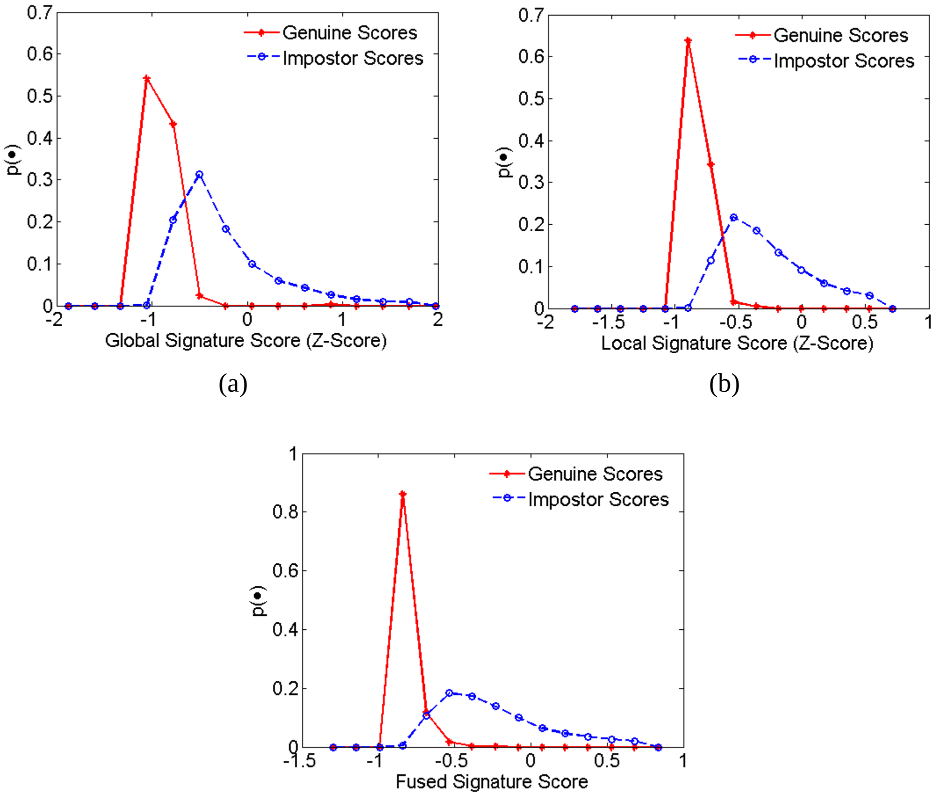
APPENDIX A

A.1 Signature Fusion for the Multimodal Database



(a) (b)

Figure A.1: Distribution of genuine and impostor signature scores from one trial of testing on the multimodal database; (a) Global (distance score) and (b) Local (distance score).



(a)

(b)

(c)

Figure A.2: Distribution of genuine and impostor signature scores from one trial of testing on the multimodal database after performing z-score normalization and sum rule fusion using weights of 0.1 and 0.9 for the global and local systems respectively; (a) Global (distance score), (b) Local (distance score) and (c) Fused scores using the sum rule.

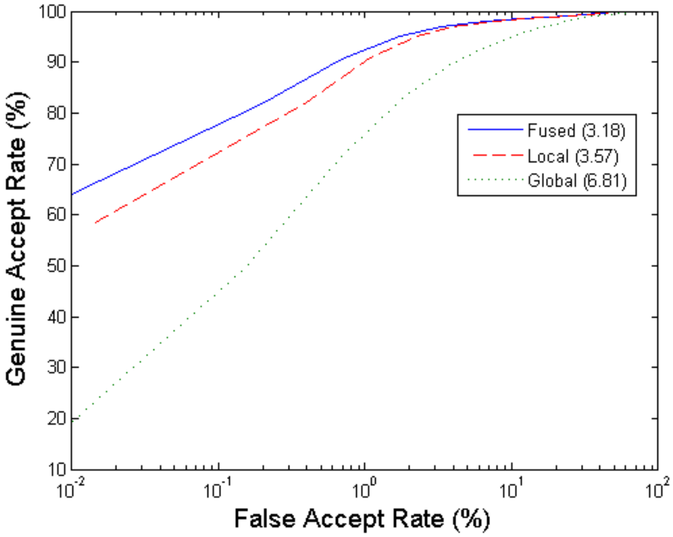


Figure A.3: Receiver operating characteristic curves for the global, local, and fused signature scores. The equal error rate (%) of each system is displayed in parenthesis in the legend.

BIBLIOGRAPHY

[1] A^4 Health Systems Electronic Medical Record Solutions. http://www.a4healthsystems.com/.

[2] Blue Cross Blue Shield of Rhode Island. https://www.bcbsri.com.

[3] HealthHighway: The Healthcare Advantage. http://www.healthhighway.com.

[4] Inova Health System. http://www.inova.org.

[5] International Biometric Industry Association. http://www.ibia.org.

[6] Protecting the Privacy of Patients' Health Information. http://www.hhs.gov/news/facts/privacy.html.

[7] Sharp: San Diego's Health Care Leader. http://www.sharp.com.

[8] University of South Alabama Health System. http://www.southalabama.edu/usahealthsystem/.

[9] George J. Annas. *The Rights of Patients*. Southern Illinois University Press, Carbondale, Illinois, 2004.

[10] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

[11] J. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85:1436–1462, 1997.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[13] Hao Feng and Chan Choong Wah. Online signature verification using a new extreme points warping technique. *Pattern Recognition Letters*, 24:2943–2951, 2003.

[14] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Penalba, J. Ortega-Garcia, and D. Maltoni. An on-line signature verification system based of fusion of local and global information. *to appear in AVBPA*, 2005.

[15] Friederike Griess. On-line signature verification. Master's thesis, Michigan State University, 2000.

[16] D'Arcy Guerin Gue. The HIPAA security rule (NPRM): Overview. http://www.hipaadvisory.com/regs/securityoverview.htm.

[17] S. Hangai, S. Yamanaka, and T. Hamamoto. Writer verification using altitude and direction of pen movement. *Proceedings of the International Conference on Pattern Recognition*, 3:483–486, 2000.

[18] Daniel L. Hartl and Elizabeth W. Jones. *Essential Genetics*. Jones and Bartlett Publishers, 2002.

[19] N. M. Herbst and C. N. Liu. Automatic signature verification based on accelerometry. *IBM Journal Of Research And Development*, 21:245–253, 1977.

[20] A. Jain and A. Ross. Learning user-specific parameters in a multibiometric system. *Proceedings of the International Conference on Image Processing*, pages 57–60, 2002.

[21] A. K. Jain, Friederike D. Griess, and Scott D. Connell. On-line Signature Verification. *Pattern Recognition*, 35(12):2963–2972, December 2002.

[22] A. K. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *to appear in Pattern Recognition*, 2005.

[23] A. K. Jain and A. Ross. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115–2125, September 2003.

[24] A. K. Jain and A. Ross. Multibiometric systems. *Communications of the ACM*, 47(1):34–40, January 2004. Special Issue on Multimodal Interfaces.

[25] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge MA, 1997.

[26] M. Kam, K. Gummadidala, G. Fielding, and R. Conn. Signature authentication by forensic document examiners. *Journal of Forensic Sciences*, 46:884–888, 2001.

[27] A. Kholmatov and B. Yanikoglu. Biometric authentication using online signatures. *ISCIS*, 3280:373–380, 2004.

[28] Alisher Kholmatov. Biometric identity verification using on-line & off-line signature verification. Master's thesis, Sabanci University, 2003.

[29] Chan F. Lam and David Kamins. Signature recognition through spectral analysis. *Pattern Recognition*, 22:39–44, 1989.

[30] Luan Ling Lee. Neural approaches for human signature verification. *Proceedings of the Third International Conference on Signal Processing*, pages 1346–1349, 1996.

[31] C. N. Liu, N. M. Herbst, and N. J. Anthony. Automatic signature verification: System description and field test results. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:35–38, 1979.

[32] Xiaoguang Lu. Image analysis for face recognition a brief survey. Personal Notes, May 2003.

[33] Paul Miller. Analyzing genetic discrimination in the workplace. *Human Genome News*, 12(1-2), February 2002.

[34] Rejean Plamondon and Guy Lorette. Automatic signature verification and writer identification - the state of the art. *Pattern Recognition*, 22:107–131, 1989.

[35] S. Prabhakar, S. Pankanti, and A.K. Jain. Biometric recognition: Security & privacy concerns. *IEEE Security & Privacy Magazine*, 1(2):33–42, March-April 2003.

[36] D. Reynolds. The effect of handset variability on speaker recognition performance: Experiments on the switchboard corpus. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1:113–116, 1996.

[37] D. Reynolds and R. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3:72–83, 1995.

[38] A. Ross and A. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24:2115–2125, 2003.

[39] A. Ross and A. Jain. Multimodal biometrics: An overview. *Proceedings of the 12th European Signal Processing Conference,* pages 1221–1224, 2004.

[40] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3), July 1979.

[41] Jianxin Yan, Alan Blackwell, Ross Anderson, and Alasdair Grant. The memorability and security of passwords - some empirical results. Technical report, University of Cambridge, 2000.

[42] L. Yang, B. K. Widjaja, and R. Prasad. Application of hidden markov models for signature verification. *Pattern Recognition*, 28:161–170, 1995.

[43] D. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. Svc2004: First international signature verification competition. *Proceedings of the International Conference on Biometric Authentication*, 2004.