Biometric Identity Verification Using On-Line & Off-Line Signature Verification

by

Alisher Anatolyevich Kholmatov

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University

Spring 2003

Biometric Identity Verification Using On-Line & Off-Line Signature Verification

APPROVED BY

Assist. Prof. Ayşe Berrin Yanıkoğlu .....
(Thesis Supervisor)

Prof. Aytül Erçil .....

Assist. Prof. Hakan Erdoğan .....

DATE OF APPROVAL: .....

to my family & my country

Acknowledgments

My sincerest thanks go to Professor Berrin Yanıkoğlu for her dedication to her students and patience in assisting me with this thesis. I appreciate her valuable advice and efforts offered during the course of my studies.

I would also like to thank my jury members, Prof. Aytül Erçil and Dr. Hakan Erdoğan, for their equally valuable support generously given during the writing of my thesis.

Special thanks go to my housemate Mansoor Naseer and my friends Thomas Bechteler and Mustafa Parlak. I appreciate their friendship and sympathetic help which made my life easier and more pleasant during graduate studies.

My colleagues and friends Zerrin Işık and İlknur Durgar receive my hearfelt thanks for their valuable friendship and discussions which facilitated my writing.

Lastly, I would like to thank my parents for their enormous encouragement and assistance, for without them, this work would not have been possible.

Abstract

Biometrics is the utilization of biological characteristics (face, iris, fingerprint) or behavioral traits (signature, voice) for identity verification of an individual. Biometric authentication is gaining popularity as a more trustable alternative to password-based security systems as it is relatively hard to be forgotten, stolen, or guessed.

Signature is a behavioral biometric: it is not based on the physical properties, such as fingerprint or face, of the individual, but behavioral ones. As such, one's signature may change over time and it is not nearly as unique or difficult to forge as iris patterns or fingerprints, however signature's widespread acceptance by the public, make it more suitable for certain lower-security authentication needs. Signature verification is split into two according to the available data in the input. Off-line signature verification takes as input the image of a signature and is useful in automatic verification of signatures found on bank checks and documents. On-line signature verification uses signatures that are captured by pressure-sensitive tablets and could be used in real time applications like credit card transactions or resource accesses.

In this work we present two complete systems for on-line and off-line signature verification. During registration to either of the systems the user has to submit a number of reference signatures which are cross aligned to extract statistics describing the variation in the user's signatures. Both systems have similar verification methodology and differ only in data acquisition and feature extraction modules. A test signature's authenticity is established by first aligning it with each reference signature of the claimed user, resulting in a number of dissimilarity scores: distances to nearest, farthest and template reference signatures. In previous systems, only one of these distances, typically the distance to the nearest reference signature or the distance to a template signature, was chosen, in an ad-hoc manner, to classify the signature as genuine or forgery. Here we propose a method to utilize all of these distances, treating them as features in a two-class classification problem, using standard pattern classification techniques. The distances are first normalized, resulting in a

three dimensional space where genuine and forgery signature distributions are well separated. We experimented with the Bayes classifier, Support Vector Machines, and a linear classifier used in conjunction with Principal Component Analysis, to classify a given signature into one of the two classes (forgery or genuine).

Test data sets of 620 on-line and 100 off-line signatures were constructed to evaluate performances of the two systems. Since it is very difficult to obtain real forgeries, we obtained skilled forgeries which are supplied by forgers who had access to signature data to practice before forging. The online system has a 1.4% error in rejecting forgeries, while rejecting only 1.3% of genuine signatures. As an offline signature is easier to forge, the offline system's performance is lower: a 25% error in rejecting forgery signatures and 20% error in rejecting genuine signatures. The results for the online system show significant improvement over the state-of-the-art results, and the results for the offline system are comparable with the performance of experienced human examiners.

Özet

Biometrik doğrulama insanın kişisel özelliklerini (parmak izi, yüz, iris, ses gibi) kullanarak gerçekleştirilen kimlik doğrulama yöntemidir. Günümüz teknolojisinin getirdiği olanaklarla önemi gün geçtikçe artan biometrik doğrulama, kart veya parola tabanlı güvenlik sistemlerine göre daha pratik (parola hatırlama, kart kaybetme ve çaldırma sorunları yok), aynı zamanda daha güvenlidir (örn. bir parolayı tahmin etmek bir parmak izini taklit etmekten daha kolaydır). İmza kişinin fiziksel özelliklerine bağlı olmayan, davranışsal bir biometriktir, bundan dolayı imza zamanla değişebilir ve parmakizi veya iris kadar özebir değildir. Göz irisi veya parmakizi gibi biometrikler kişiye özgü olmalarına karşın, suçlular ile ilişkilendirildikleri ve kişi hakkında sağlık gibi konularda istenmeyen bilgileri açığa çıkardıkları için, bu sistemleri kullanmaya başlayan ülkelerde toplum tarafından kolaylıkla kabul görmemişlerdir. Öte yandan imza, günümüzde hemen her ortamda kimlik doğrulama işlemleri için gerekli bir bilgi olarak görülmektedir.

İmza doğrulama statik (off-line) veya dinamik (on-line) imza doğrulama şeklinde iki ana konu olarak değerlendirilmektedir. Kağıt üzerindeki statik bir imzadan, tarama yoluyla sadece imzanın şeklini içeren bir imge elde edilmesine karşın, dokunmaya hassas tabletlere atılan dinamik imzalarda hem imzanın şekli, hem de dinamik özellikleri (hızı, kaç darbede atıldığı, kalemin ne kadar bastırıldığı gibi) elde edilebilir. Statik bir imzanın kopyalanması elde bir örnek varsa oldukça kolay olmasına karşın, dinamik özellikler imzayı daha kişiye özgü kılar ve taklit edilmesini zorlaştırır. Yine de her iki imza türüne dayalı doğrulama sistemlerinin kullanım alanları farklıdır: mesela statik imza doğrulayıcı bir sistem banka çeklerindeki sahteciliklerin yakalanmasında kullanılırken, dinamik imza doğrulama sistemleri özellikle kredi kartındaki sahteciliklerin yakalanmasında kullanılmaktadır. Dinamik imza doğrulama sistemleri ayrıca bina girişlerinde, eliçi ve avuçiçi bilgisayarlarındaki bilgilerin korunmasında kullanılmaktadır.

Bu çalışmada iki ayrı imza türüne dayalı (statik ve dinamik) iki farklı imza doğrulama sistemi sunulmaktadır. Her iki sistemde de kullanıcı bir kaç referans imza

vererek sisteme kaydolur. Bu referans imzalarından, kişinin imzalarının özelliklerini ve değişkenliğini karakterize eden öznitelikler çıkarılır ve sistemde bu kullanıcıya özgü değerler olarak saklanır. Her iki sistemin girdi olarak kabul ettikleri imza türleri ve imzalardan çıkarılan öznitelikler farklı olmalarına rağmen, sistemler aynı doğrulama yöntemine dayanmaktadırlar: herhangi bir imza doğrulanacağı zaman, bu imza iddia edilen kişinin bütün referans imzalarıyla karşılaştırılır ve test edilen imzanın referans imzalarına uzaklığı (farklılığı) hesaplanır. Herhangi iki imza arasındaki farklılık, farklı uzunluklardaki iki dizinin, linear olmayan bir değişimle gelebilecekleri en benzer hallerin uzaklığını hesaplamak için kullanılan "Dynamic Time Warping" algoritması ile bulunur. Daha önce geliştirilmiş imza doğrulama sistemlerinde, bu işlemin sonucunda elde edilen minimum uzaklık (test imzasının en yakın referans imzasına uzaklığı) veya test imzasının şablon referans imzasına uzaklığı, bu kişiye ait ortalama değerlerle karşılaştırılarak, imzanın gerçek mi, taklit mi olduğuna buluşsal yöntemlerle karar verilmekteydi. Önerdiğimiz doğrulama yönteminde bahsi geçen uzaklıklar kendilerine karşılık gelen referans imzalar arasındaki ortalama uzaklıklarla normalize edilerek, sahte ve gerçek imzaların birbirinden ayrık oldukları öznitelik uzayı oluşturmaktadırlar. Çalışmamızda imzalardan çıkarılan üç boyutlu öznitelik vektörleri Bayes sınıflandırıcı, Destekçi Vektör Makinesi, ve Linear sınıflandırıcı kullanarak imzaların sahte olup olmadığını tespit etmek için kullanılmışlardır.

Sistemleri denemek için 100 ayrı kişiden toplam 620 dinamik ve 20 kişiden toplam 100 statik deneme imzası (gerçek ve sahte) toplanmıştır. Gerçek taklit imzaları elde etmek zor olduğu için, taklit edeceği imzanın şeklini ve mümkünse imzalama haraketlerini görebilen taklitçilerden nitelikli sahte imzalar alınmıştır. Dinamik imza doğrulama sistemi gerçek imzaların %1.4'ünü yanlışlıkla redederken, sahte imzaların sadece %1.3'ü yanlışlıkla kabul etmiştir. Statik imzayı taklit etmek daha kolay olduğu için, statik imza doğrulama sistemi sahte imzaların %25'ini yanlışlıkla kabul ederken, gerçek imzaların %20'sini yanlışlıkla redetmiştir. Önerilen dinamik doğrulama sistemi var olan sistemlerden daha üstün performans sergilerken, statik doğrulama sistemimizden de bu konudaki uzman kişilerin başarısıyla kıyaslanabilir performans elde edilmiştir.

Table of Contents

List of Figures

List of Tables

Biometric Identity Verification Using On-Line & Off-Line Signature Verification

by

Alisher Anatolyevich Kholmatov

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University

Spring 2003

Biometric Identity Verification Using On-Line & Off-Line Signature Verification

APPROVED BY

Assist. Prof. Ayşe Berrin Yanıkoğlu .....
(Thesis Supervisor)

Prof. Aytül Erçil .....

Assist. Prof. Hakan Erdoğan .....

DATE OF APPROVAL: .....

to my family & my country

Acknowledgments

My sincerest thanks go to Professor Berrin Yanıkoğlu for her dedication to her students and patience in assisting me with this thesis. I appreciate her valuable advice and efforts offered during the course of my studies.

I would also like to thank my jury members, Prof. Aytül Erçil and Dr. Hakan Erdoğan, for their equally valuable support generously given during the writing of my thesis.

Special thanks go to my housemate Mansoor Naseer and my friends Thomas Bechteler and Mustafa Parlak. I appreciate their friendship and sympathetic help which made my life easier and more pleasant during graduate studies.

My colleagues and friends Zerrin Işık and İlknur Durgar receive my hearfelt thanks for their valuable friendship and discussions which facilitated my writing.

Lastly, I would like to thank my parents for their enormous encouragement and assistance, for without them, this work would not have been possible.

Abstract

Biometrics is the utilization of biological characteristics (face, iris, fingerprint) or behavioral traits (signature, voice) for identity verification of an individual. Biometric authentication is gaining popularity as a more trustable alternative to password-based security systems as it is relatively hard to be forgotten, stolen, or guessed.

Signature is a behavioral biometric: it is not based on the physical properties, such as fingerprint or face, of the individual, but behavioral ones. As such, one's signature may change over time and it is not nearly as unique or difficult to forge as iris patterns or fingerprints, however signature's widespread acceptance by the public, make it more suitable for certain lower-security authentication needs. Signature verification is split into two according to the available data in the input. Off-line signature verification takes as input the image of a signature and is useful in automatic verification of signatures found on bank checks and documents. On-line signature verification uses signatures that are captured by pressure-sensitive tablets and could be used in real time applications like credit card transactions or resource accesses.

In this work we present two complete systems for on-line and off-line signature verification. During registration to either of the systems the user has to submit a number of reference signatures which are cross aligned to extract statistics describing the variation in the user's signatures. Both systems have similar verification methodology and differ only in data acquisition and feature extraction modules. A test signature's authenticity is established by first aligning it with each reference signature of the claimed user, resulting in a number of dissimilarity scores: distances to nearest, farthest and template reference signatures. In previous systems, only one of these distances, typically the distance to the nearest reference signature or the distance to a template signature, was chosen, in an ad-hoc manner, to classify the signature as genuine or forgery. Here we propose a method to utilize all of these distances, treating them as features in a two-class classification problem, using standard pattern classification techniques. The distances are first normalized, resulting in a

three dimensional space where genuine and forgery signature distributions are well separated. We experimented with the Bayes classifier, Support Vector Machines, and a linear classifier used in conjunction with Principal Component Analysis, to classify a given signature into one of the two classes (forgery or genuine).

Test data sets of 620 on-line and 100 off-line signatures were constructed to evaluate performances of the two systems. Since it is very difficult to obtain real forgeries, we obtained skilled forgeries which are supplied by forgers who had access to signature data to practice before forging. The online system has a 1.4% error in rejecting forgeries, while rejecting only 1.3% of genuine signatures. As an offline signature is easier to forge, the offline system's performance is lower: a 25% error in rejecting forgery signatures and 20% error in rejecting genuine signatures. The results for the online system show significant improvement over the state-of-the-art results, and the results for the offline system are comparable with the performance of experienced human examiners.

Özet

Biometrik doğrulama insanın kişisel özelliklerini (parmak izi, yüz, iris, ses gibi) kullanarak gerçekleştirilen kimlik doğrulama yöntemidir. Günümüz teknolojisinin getirdiği olanaklarla önemi gün geçtikçe artan biometrik doğrulama, kart veya parola tabanlı güvenlik sistemlerine göre daha pratik (parola hatırlama, kart kaybetme ve çaldırma sorunları yok), aynı zamanda daha güvenlidir (örn. bir parolayı tahmin etmek bir parmak izini taklit etmekten daha kolaydır). İmza kişinin fiziksel özelliklerine bağlı olmayan, davranışsal bir biometriktir, bundan dolayı imza zamanla değişebilir ve parmakizi veya iris kadar özebir değildir. Göz irisi veya parmakizi gibi biometrikler kişiye özgü olmalarına karşın, suçlular ile ilişkilendirildikleri ve kişi hakkında sağlık gibi konularda istenmeyen bilgileri açığa çıkardıkları için, bu sistemleri kullanmaya başlayan ülkelerde toplum tarafından kolaylıkla kabul görmemişlerdir. Öte yandan imza, günümüzde hemen her ortamda kimlik doğrulama işlemleri için gerekli bir bilgi olarak görülmektedir.

İmza doğrulama statik (off-line) veya dinamik (on-line) imza doğrulama şeklinde iki ana konu olarak değerlendirilmektedir. Kağıt üzerindeki statik bir imzadan, tarama yoluyla sadece imzanın şeklini içeren bir imge elde edilmesine karşın, dokunmaya hassas tabletlere atılan dinamik imzalarda hem imzanın şekli, hem de dinamik özellikleri (hızı, kaç darbede atıldığı, kalemin ne kadar bastırıldığı gibi) elde edilebilir. Statik bir imzanın kopyalanması elde bir örnek varsa oldukça kolay olmasına karşın, dinamik özellikler imzayı daha kişiye özgü kılar ve taklit edilmesini zorlaştırır. Yine de her iki imza türüne dayalı doğrulama sistemlerinin kullanım alanları farklıdır: mesela statik imza doğrulayıcı bir sistem banka çeklerindeki sahteciliklerin yakalanmasında kullanılırken, dinamik imza doğrulama sistemleri özellikle kredi kartındaki sahteciliklerin yakalanmasında kullanılmaktadır. Dinamik imza doğrulama sistemleri ayrıca bina girişlerinde, eliçi ve avuçiçi bilgisayarlarındaki bilgilerin korunmasında kullanılmaktadır.

Bu çalışmada iki ayrı imza türüne dayalı (statik ve dinamik) iki farklı imza doğrulama sistemi sunulmaktadır. Her iki sistemde de kullanıcı bir kaç referans imza

vererek sisteme kaydolur. Bu referans imzalarından, kişinin imzalarının özelliklerini ve değişkenliğini karakterize eden öznitelikler çıkarılır ve sistemde bu kullanıcıya özgü değerler olarak saklanır. Her iki sistemin girdi olarak kabul ettikleri imza türleri ve imzalardan çıkarılan öznitelikler farklı olmalarına rağmen, sistemler aynı doğrulama yöntemine dayanmaktadırlar: herhangi bir imza doğrulanacağı zaman, bu imza iddia edilen kişinin bütün referans imzalarıyla karşılaştırılır ve test edilen imzanın referans imzalarına uzaklığı (farklılığı) hesaplanır. Herhangi iki imza arasındaki farklılık, farklı uzunluklardaki iki dizinin, linear olmayan bir değişimle gelebilecekleri en benzer hallerin uzaklığını hesaplamak için kullanılan "Dynamic Time Warping" algoritması ile bulunur. Daha önce geliştirilmiş imza doğrulama sistemlerinde, bu işlemin sonucunda elde edilen minimum uzaklık (test imzasının en yakın referans imzasına uzaklığı) veya test imzasının şablon referans imzasına uzaklığı, bu kişiye ait ortalama değerlerle karşılaştırılarak, imzanın gerçek mi, taklit mi olduğuna buluşsal yöntemlerle karar verilmekteydi. Önerdiğimiz doğrulama yönteminde bahsi geçen uzaklıklar kendilerine karşılık gelen referans imzalar arasındaki ortalama uzaklıklarla normalize edilerek, sahte ve gerçek imzaların birbirinden ayrık oldukları öznitelik uzayı oluşturmaktadırlar. Çalışmamızda imzalardan çıkarılan üç boyutlu öznitelik vektörleri Bayes sınıflandırıcı, Destekçi Vektör Makinesi, ve Linear sınıflandırıcı kullanarak imzaların sahte olup olmadığını tespit etmek için kullanılmışlardır.

Sistemleri denemek için 100 ayrı kişiden toplam 620 dinamik ve 20 kişiden toplam 100 statik deneme imzası (gerçek ve sahte) toplanmıştır. Gerçek taklit imzaları elde etmek zor olduğu için, taklit edeceği imzanın şeklini ve mümkünse imzalama haraketlerini görebilen taklitçilerden nitelikli sahte imzalar alınmıştır. Dinamik imza doğrulama sistemi gerçek imzaların %1.4'ünü yanlışlıkla redederken, sahte imzaların sadece %1.3'ü yanlışlıkla kabul etmiştir. Statik imzayı taklit etmek daha kolay olduğu için, statik imza doğrulama sistemi sahte imzaların %25'ini yanlışlıkla kabul ederken, gerçek imzaların %20'sini yanlışlıkla redetmiştir. Önerilen dinamik doğrulama sistemi var olan sistemlerden daha üstün performans sergilerken, statik doğrulama sistemimizden de bu konudaki uzman kişilerin başarısıyla kıyaslanabilir performans elde edilmiştir.

Table of Contents

List of Figures

List of Tables

Chapter 1

Biometric Authentication

Automatically verifying someone's identity by his face, iris or fingerprint is no longer science fiction, but rather it became a daily routine authentication procedure in many places. Biometrics is the utilization of physiological characteristics (face, iris, fingerprint) or behavioral traits (signature, voice) for identity verification of an individual, though the complete list of characteristics is much longer. Biometric authentication is gaining popularity as a more trustable alternative to password-based security systems, since it is almost impossible to steal, copy, or guess biometric properties. Furthermore, one can forget his password, whereas forgetting is even not an issue for biometric properties.

While looking for a proper biometric to be used in a particular application, the following criteria are important: i) uniqueness, ii) whether it is hard to be copied or stolen, iii) acceptability by the public, iv) and the cost to employ that particular biometric data.

Signature is a behavioral biometric: it is not based on physiological properties of the individual, such as fingerprint or face, but behavioral ones. As such, one's signature may change over time and it is not nearly as unique or difficult to forge as iris patterns or fingerprints, however signature's widespread acceptance by the public, make it more suitable for certain lower-security authentication needs. For instance, MasterCard estimates a $450 million loss each year due to credit card fraud, likewise some billions of dollars being lost because of fraudulent encashment of checks. Reliable automatic signature verification could be a proper solution to reduce such losses since handwritten signatures are already involved in the credit card transactions and bank checks encashment.

Signature verification is split into two according to the available data in the input. Offline (static) signature verification takes as input the image of a signature and is useful in automatic verification of signatures found on bank checks and documents. Online (dynamic) signature verification uses signatures that are captured by pressure-sensitive tablets that extract dynamic properties of a signature in addition to its shape, and can be used in real time applications like credit card transactions, protection of small personal devices (e.g. PDA, laptop), authorization of computer users for accessing sensitive data or programs, and authentication of individuals for access to physical devices or buildings.

Signatures in off-line systems usually may have noise, due to scanning hardware or paper background, and contain less discriminative information since only the image of the signature is the input to the system. While genuine signatures of the same person may slightly vary, the differences between a forgery and a genuine signatures may be imperceptible, which make automatic off-line signature verification be a very challenging pattern recognition problem. Besides, the difference in pen widths and unpredictable change in signature's aspect ratio are other difficulties of the problem. Worth to notice is the fact that even professional forensic examiners perform at about 70% of correct signature classification rate (genuine or forgery). On-line signatures are more unique and difficult to forge than their counterparts are, since in addition to the shape information, dynamic features like speed, pressure, and capture time of each point on the signature trajectory are available to be involved in the classification. In other words, on-line signatures have an extra dimension, which is not available for the off-line signatures. As a result, on-line signature verification is more reliable than the off-line.

Figure 1.1 summarizes the task to be solved by a signature verification system: given a test signature and a claimed ID, either accept a user as the identity owner or deny him based on a dissimilarity degree between the test and reference set signatures. In either of the signature verification systems, the users are first enrolled by providing reference signature samples. When a user presents a test signature and claims to be a particular individual, the test signature is compared with reference set signatures of the claimed identity. If the dissimilarity between the test and reference set signatures is above a certain threshold, the user is rejected, otherwise accepted.

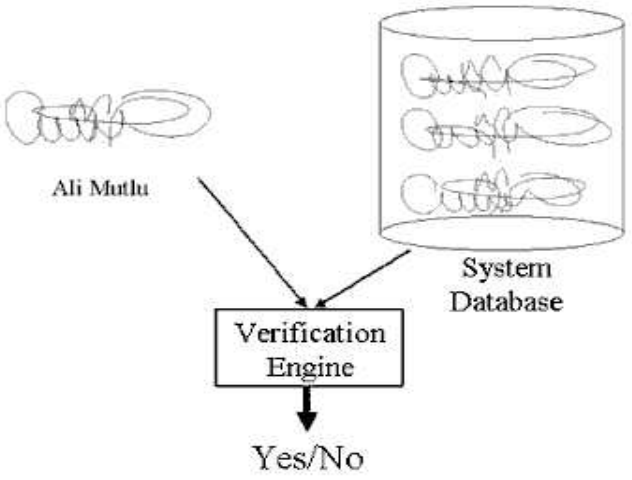

Figure 1.1: The task of an automatic signature verification system.

The dissimilarity between two signatures can be established in two ways: if each time a signature is presented to the system, equal number of features are being extracted from that signature, some sort of distance (ex. Euclidian distance) can be used to compare these two signatures. In this type of comparison, global features which describe the signature as a whole, are used. Systems using only global features are generally fast but have low performance. The second alternative is to make a point-by-point comparison, where the so called local features, pertaining to particular points on the signature trajectory, are used. Since even signatures signed by the same person may vary in length (implying feature vectors of different length), methods which are able to non-linearly associate vectors of different lengths, such as Dynamic Time Warping (DTW) or Hidden Markov Models (HMM) are used.

In evaluating the performance of a signature verification system, there are two important factors: the false rejection rate (FRR) of genuine signatures and the false acceptance rate (FAR) of forgery signatures. As these two are inversely related, lowering one often results in increasing the other. Hence, it is common to talk about the equal error rate (EER) which is the point where FAR equals FRR. Since obtaining actual forgeries is difficult, two forgery types have been defined: A *skilled* forgery is signed by a person who has had access to a genuine signature for practice. A *random* or *zero-effort forgery* is signed without having any information about the signature, or even the name, of the person whose signature is forged. State of the art performance of the available on-line signature verification algorithms lies between

1% and 10% equal error rate, while off-line verification performance is still between 70% and 80% equal error rate. Unfortunately no public signature database of either type is available, which makes it difficult to compare existing signature verification systems.

Chapter 2

On-Line Signature Verification

This chapter describes our on-line signature verification system. In Section 2.1 we make a literature overview of existing methods for the on-line signature verification. In Section 2.2, there is an overview of the system and its main modules. Section 2.3 covers the data acquisition process and the commercially available hardware used for that purpose. Section 2.4 is on commonly used preprocessing techniques. Feature extraction and dissimilarity comparison between two signatures are covered in Sections 2.5 and 2.6, respectively. Enrollment to the system and verification phases are described in Sections 2.7 and 2.8, respectively. Performance results of the system are presented in Section 2.9. Finally, a summary of proposed system is done in Section 2.10.

2.1 Literature Overview

Advances in technology and relatively cheap data acquisition devices triggered the use of on-line signature verification in many real time applications, such as credit card transactions, document flow applications, and identity authentication prior to access of sensitive resources. There have been several studies on on-line signature verification problem. On-line signature verification systems differ on various issues, such as data acquisition, preprocessing, and dissimilarity calculation. These issues and some of the existing methods are discussed in this section.

Most commonly used on-line signature acquisition devices are pressure sensitive tablets with or without visual feedback. Smart pens capable of measuring forces at the pen-tip, exerted in three direction, are also widely used in signature verifica-

tion systems. Special hand gloves with sensors for detecting finger bend and hand position and orientation [33], and a CCD camera based [19] approaches were also in signature acquisition; however, due to their cost and impracticality, such devices couldn't find place in real systems. Depending on the device used, fair amount of preprocessing may be applied to a signature data prior to the feature extraction phase [13, 21]. We discuss commonly used preprocessing techniques in Section 2.4.

In addition to the trajectory coordinates, behavioral characteristics, such as pressure at pen tip, acceleration, and pen tilt, can be captured during the signing session, depending on the device used. Using these characteristics more than 40 features [35] have been used for signature verification. Features can be classified in two types: global and local. Global features are features related to the signature as a whole; for instance the signing speed, signature bounding box, and Fourier descriptors of the signature's trajectory. Local features correspond to a specific sample point along the trajectory of the signature. Examples of local features include distance and curvature change between successive points on the signature trajectory. Some researchers tried to find a set of robust and discriminative features for signature verification purposes [5, 13, 26], however the sets were selected experimentally and may only be applicable for particular verification methods. Genetic Algorithms were also used to find the most useful set of features [36].

Due to behavioral changes of a writer, two signatures signed by the same person may have different trajectory lengths (hence feature vectors of differing lengths). Therefore, straight forward methods, such as the Euclidian distance or autocorrelation, are not very useful in calculation of the dissimilarity value between two signatures. To overcome the problem, methods which can non-linearly relate vectors of different length are commonly used. For instance, dynamic time warping algorithm with some sort of the Euclidian distance [13, 15, 21, 23] and Hidden Markov Models [5, 26] are commonly used in aligning two signatures.

Generally in previous systems, between 3 and 20 reference signatures are taken during the user enrollment. Template generation for the reference set signatures is generally accomplished by simply selecting one or more of the sample signatures as templates [21, 23]. Various thresholds are used in deciding whether the dissimilarity between the test signature and the reference and/or template signatures is accept-

able. Two types of threshold selections were reported: writer dependent and writer independent thresholds [13]. In writer dependent scenario, thresholds are calculated for each user individually, whereas in writer independent one, a global threshold for all the writers is set empirically during the validation phase of the system.

State of the art performance of the existing on-line signature verification algorithms lies between 1% and 10% equal error rate. However lack of publicly available signature database and difficulties in obtaining skilled forgeries make it difficult to do a comprehensive comparison between existing on-line signature verification methods.

Previously Proposed Methods

Jain et al. [13] used pressure sensitive tablet to capture signatures. After a fair amount of preprocessing (resampling, smoothing, and size normalization), several local features were extracted: x,y coordinate differences between two consecutive points, curvature, gray values in 9x9 neighborhood, absolute and relative speeds, etc. Number of signature strokes was the only extracted global feature, which was later incorporated to the overall dissimilarity value. Dynamic programming algorithm was applied to align two signatures. The overall dissimilarity value between a test and a template signatures was then calculated by linearly incorporating the alignment score, the difference of stroke numbers between the signatures, and the normalization factor. Three different criteria were investigated to authenticate the test signature: the minimum, the maximum, and the average dissimilarity values to the reference set signatures. Finally, the common and the writer-dependent thresholds were separately used to classify the signature as genuine or forgery. System was tested using a test data set of 1232 genuine and 60 skilled forgery signatures, captured from 102 individuals. In addition to that, system was also tested against random forgeries, where authentic signatures of enrolled writers served as random forgeries to each other. Jain et al. reported best results using minimum dissimilarity criterion and writer-dependent thresholds, where the system performance was a 2.8% false accept rate and 1.6% false reject rate using only random forgeries. Using common threshold yielded a 3.3% false reject rate and a 2.7% false accept rate again using only random forgery signatures.

Nalwa in his work [21] claims that the behavioral characteristics of a signature are not as consistent as it's shape information. He summarizes his algorithm in three phases: normalization, description, and comparison. Normalization was used to make the algorithm invariant to changes in signature's orientation (rotation) and aspect ratio (size). First a polygon was fitted through the sample points of signature trajectory. Then signature was normalized with respect to rotation and aspect ratio of fitted polygon. The jitter, the aspect ratio and number of strokes were extracted prior to the normalization, and kept as global features. During the description phase, five characteristic functions were derived, each describing a local feature of the signature. Features described are: the x and y coordinates relative to the center of mass, the torque and two curvature-ellipses measures derived from the moments of inertia. Each function then was normalized to have zero mean. Finally, comparison was providing the dissimilarity measure between the signature and a claimed prototype. To do so, characteristic functions were simultaneously warped against their prototypes, resulted in the overall alignment cost. The alignment cost was then considered as a global feature. The final dissimilarity measure was defined as the weighted harmonic mean of the global features. The system was tested using three different data sets of 904, 982 and 790 genuine signatures, where 59, 102 and 43 writers contributed to, respectively. Additionally, 325, 401 and 424 forgery signatures were collected. Using 6 reference signatures for the prototype creation, Nalwa reported equal error rates of 3%, 2% and 5%, for each data set respectively.

Dolfing et al. [5] used a special digitizer consisting of an LCD and orthogonal sensors for pen-tilt tracking. Using this setup, x and y coordinates, pressure at pen tip and a pen tilt in the x and y directions were captured. A signature was divided into number of segments, where segment boundaries were identified using velocity inversion criterion (i.e. v_y=0). 32 features were extracted for each segment : 13 spatial, 13 dynamic, and 6 contextual. Each signature was modeled by a single left-to-right Hidden Markov model, where loop, forward and skip transition probabilities were estimated during training. The observation probabilities were continuous Gaussian mixtures and up to four Gaussians were allowed per each state. The number of states was equal to 0.8 times number of segments of a reference signature. The model was trained using the Maximum Likelihood criterion and applying Viterbi

algorithm, followed by linear discriminant analysis. Test signature's dissimilarity calculation was based on the Viterbi algorithm, which calculates the likelihood of the signature being generated by the claimed writer's model. An adaptive threshold, which is a combination of a common offset and a writer dependent threshold, was used for accepting or rejecting a test signature. A test data set of 1530 genuine and 3000 amateur forgeries was constructed, using signatures collected from 51 individuals. Furthermore, 240 skilled forgeries were supplied by 6 professional document examiners. In average, an equal error rate of 2.45% was obtained.

Rigoll et al. [26] provided a comparison between on-line and off-line signature verification using Hidden Markov Models. Signatures used for either of the systems were from the same data set; hence while using signatures for the off-line verification system ,all dynamic features were discarded and only the image of the signature was used. Seven different feature types were empirically tested for their discriminative capabilities. Although Rigoll et al. used discrete Hidden Markov Models, they didn't mention about the structure of the models. The Viterbi algorithm was used to compute the likelihood probability of a test signature belonged to a claimed writer's model. The system was tested on very small data set: 14 writers contributed to the data set with 20 signatures each, 16 of which were used for training each writer's model, and the remaining 4 (56 total) were used for testing. As for the forgery set, 60 forgeries were supplied by 10 forgers, where 40 of them were skilled forgeries. Each feature was evaluated for it's discriminative power. Then empirically combined feature sets were tested in the same manner. The feature set of bitmap, velocity, Fourier transform and pressure features yielded the best performance results of 1% equal error for the on-line system. For the off-line case an equal error rate of 1.9% was obtained. Although good performance results are reported for these systems, the data sets are too small to give reliable performance numbers.

2.2 General System Overview

Figure 2.1 depicts a high level representation of the proposed on-line signature verification system. Data acquisition module is responsible for capturing signature data during the signing session. Profile generator creates a profile, based on the infor-

mation extracted from the reference signatures of the user, which is then stored in the system database. Verification engine is responsible for the verification of a given test signature, based on the dissimilarity between the test and the reference set signatures.

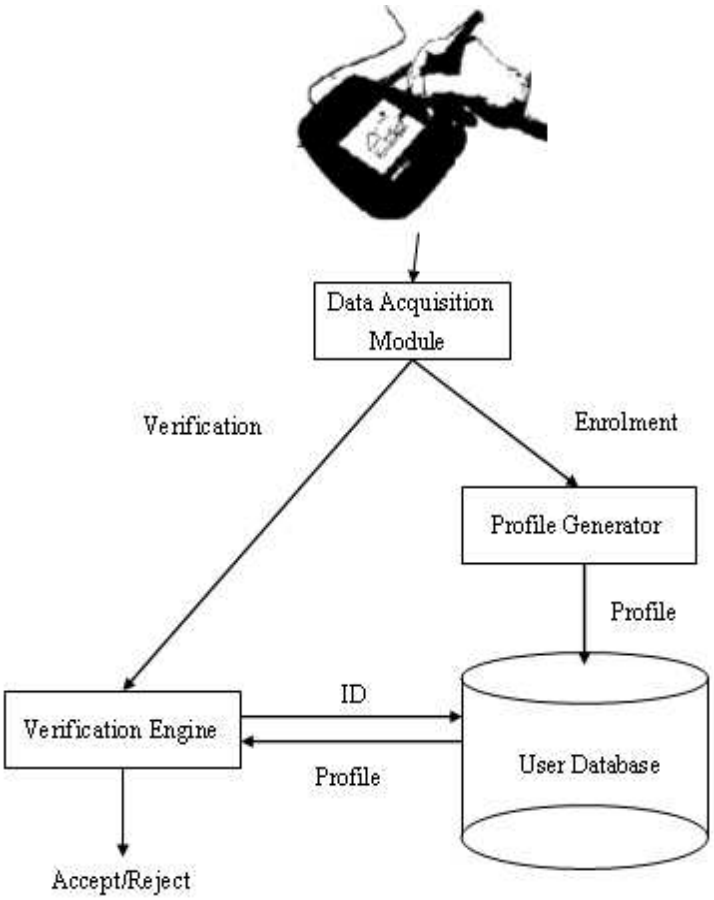

Figure 2.1: High level representation of the proposed on-line signature verification system.

An on-line signature can be viewed as a function of time. This fact makes it easier to derive the signing characteristics for a particular user, as explained in Section 2.5. Due to the same fact, on-line signature verification systems are more reliable compared to off-line signature verification systems.

Figure 2.2 depicts a sample on-line signature in our database. Arrows in the figure 2.3 show signing flow of the signature which can't be easily deduced even for such simple signature (numbers indicate signing sequence of signature strokes). The

sampling points of that signature are depicted in Figure 2.4. Distances between sampling points are not even, caused by the variation of signing speed with time, which is a behavioral characteristic of a writer. Depending on the device used, behavioral characteristics, such as pressure at pen tip, acceleration, and pen tilt, can be captured during the signing session. Overview of the commercially available data acquisition hardware is presented in Section 2.3.

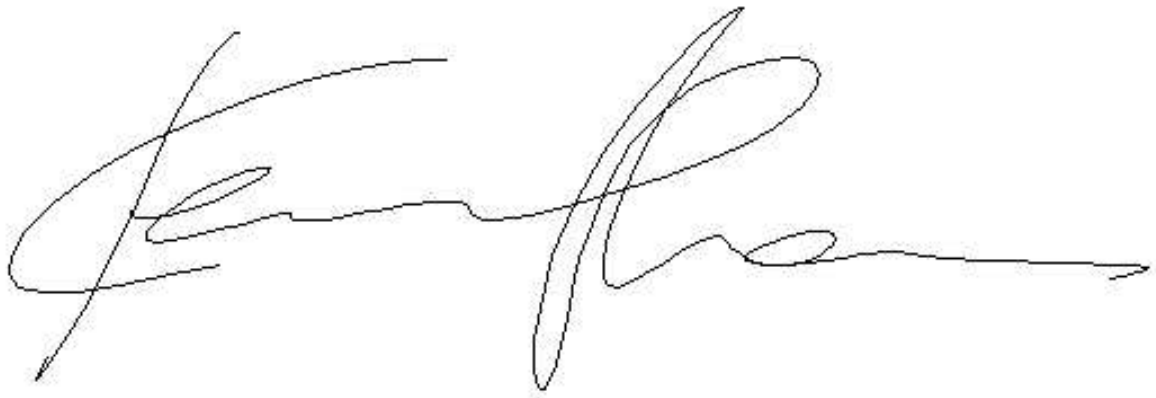

Figure 2.2: Sample on-line signature from our signature database.

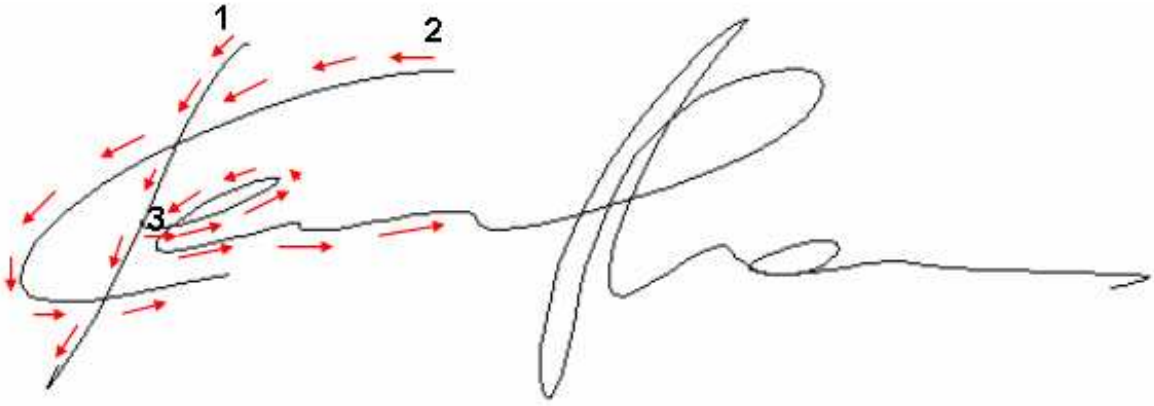

Figure 2.3: Signing flow of the sample on-line signature. Red arrows show signing flow and numbers indicate signing sequence of signature strokes.

Some of the data acquisition hardware may introduce noise and jaggedness to the signature data. Similarly, use of different acquisition devices within the same system may introduce change in signature's scale and orientation. Most commonly

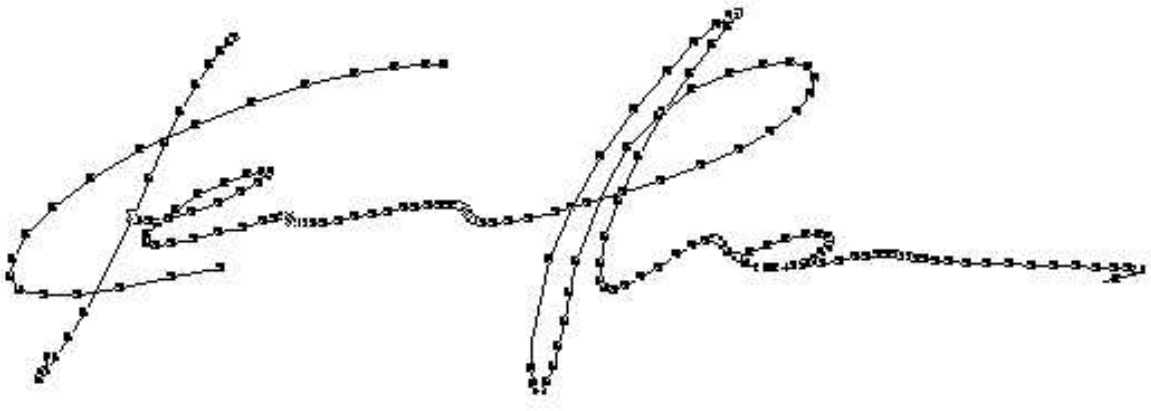

Figure 2.4: Sampling points of the example on-line signature.

used preprocessing techniques to remove such variations, along with their advantages and shortcomings are described in Section 2.4.

During enrollment the user gives a number of reference signatures which are used in creating a profile for that user in the system. The user profile contains supplied reference signatures and similarity values which describe variations within the reference signatures. Similarity between two signatures is calculated using dynamic programming algorithm, as described in Section 2.6. More detailed information about similarity values and the way they are being extracted is provided in Section 2.7.

Verification engine is used to authenticate a given (test) signature against the claimed ID. The test signature is compared with each reference signature using dynamic programming algorithm. Comparison results in a number of similarity values, which are then presented to a classifier for a final decision. We have experimented with Support Vector Machine, Bayes, and Linear classifiers. Verification process is broadly described in Section 2.8.

2.3 Data Acquisition

Digital tablets are one of the oldest types of input devices used with the computer. In the 1950s, US military used a type of digital tablet in a system developed to help in manipulation of radar images. These systems cost millions of dollars and filled entire rooms. In the 1970s, minicomputers that used CAD (Computer Aided

Design) became available. These systems used a puck to input information. The puck resembled a mouse, only it had a lens with a crosshair mounted in the front part. The puck was used on a special tablet and contained numerous buttons [32]. Pressure tablets were also available for Amiga; for instance EasyL is one of them. It had an active area of 8.5"x11" and only sensed the pressure being on or off (no variable pressure). Today pressure sensitive tablets are very common and they are a relatively cheap computer accessory. Pressure sensitive tablets, also called graphics tablets or pads, are widely used for graphics manipulation, CAD, web browsing and simply instead of a mouse.

Figure 2.5: Interlink Electronics ePad-ink pressure sensitive tablet with visual feedback.

There are some key points which determine the quality and possible application areas of a tablet: size of the active area of a pad, resolution, pressure sensitivity levels, sampling rate, and availability of visual feedback. Input device capabilities determine the quality of signature features, being extracted during the signing sessions, and directly effect performance of the systems.

Tablets are not the only possible input device for on-line signature verification systems. Digital pens or smart pens with some special sensors at pen tip are an alternative to pressure sensitive tablets. For instance, the FingerSystem's *i-pen* has an optical sensor which provides accurate and precise position of pen motion or LCI's *SmartPen* which has sensors that determine the angle and precise movements of the pen. SmartPen is also capable of reading and converting writing or voice into computer text. Yet another example is Logitech's *Digital Pen*. A comprehensive

list of the pressure sensitive tablets that are commercially available in the market, is presented in Table 2.1.

Brand & Model	*Active Area*	*Pressure Levels*	*Resolution*
Interlink ePad-ink	3"x2.20"	512	300dpi
Wacom Graphire2	3.65" x 5"	512	1016lpi
Aiptek Hyperpen 6000U	4.5"x6"	512	3048lpi
Dynalink FreeDraw	5"x3.75"	512	2540lpi
Genius EasyPen	4"x3"	-	2540lpi
Genius WizardPen	4"x3"	512	4064lpi
Genius MousePen	5.5"x4"	512	4064lpi
CalComp DrawingBoard III	12"x12"	256	2540lpi
Paradise Graphics Tablet	5"x4"	512	2048dpi
UC-Logic SuperPen 4030	4"x3"	512	1000lpi
UC-Logic SuperPen 8060	8"x6"	1024	1000lpi
Acedad Flair	5"x3.75"	512	2540lpi

Table 2.1: Pressure sensitive tablets available in the market.

We have used both Wacom's *Graphire2* pressure sensitive tablet and Interlink's *ePad-ink* with visual feedback. Both tablets are capable of sampling data at about 100 samples per second: at each sample point, the x,y coordinates of the signature's trajectory and the time stamp are recorded. Wacom's pen is featured to capture samples only during the interaction of the pen tip with the tablet. ePad-ink doesn't require special pen to be used and is capable of giving visual feedback (Figure 2.5) through a LCD screen, which gives to a signer natural feeling of signing on ordinary paper.

2.4 Preprocessing

There are some commonly done preprocessing steps, aimed to improve the verification performance of a system. These range from size normalization to smoothing of the trajectory and resampling. All of the preprocessing techniques are done at the

expense of removing some properties peculiar to the particular writer. There may be some circumstances where performing these are inevitable, such as when using noisy data acquisition devices or when there are discrepancies among the hardware devices within the system. In such cases, one should carefully choose and design the preprocessing phase of the system. Within our setup, where the hardware was one type and had a sufficient resolution, we decided to bypass preprocessing so that the timing characteristics of the writer were not discarded.

Tablets with low resolutions or low sampling rates may give signatures that have jaggedness which is commonly removed using smoothing techniques. However, tremor in the signature, which can also cause the jaggedness, may be a behavioral characteristic of a writer. Applying smoothing will remove that characteristic.

In the systems where tablets of different active areas are used, signature size normalization is a frequently used preprocessing technique. Comparing two signatures having the same shape but different sizes would result in low similarity scores, when using some of the comparison techniques, such as point-by-point comparison by applying dynamic programming algorithm. Size normalization is commonly applied to obtain scale invariance for such comparison algorithms. However, the size may be a writer dependent characteristic, i.e. writer may always sign in only large or small signatures, whereas normalization will remove it.

Modern tablets have a sampling rate of more than 100 trajectory points per second. In some of the previous methods, resampling, as a preprocessing step, was used to get rid of possibly redundant data . After successful resampling, shape related features were more reliably extracted, however this was done at an expense of loosing speed information, implicitly incorporated in the data.

2.4.1 Resampling

Due to the high sampling rate of the tablet, some sample points mark the same trajectory point, especially when the pen movement is slow. Most verification systems resample the input so as to obtain a trajectory consisting of equidistant points [13, 15, 36]. This is often done in order to remove redundant points to speed up the comparisons and to obtain a shape-based representation, removing the time dependencies. However, resampling also results in significant loss of information

since the seemingly redundant data incorporates speed characteristics of the genuine signer. It is very difficult to catch and imitate the signing dynamics of the original signature. Furthermore, a signature is considered as a ballistic movement such as handwriting or throwing a ball, and a forger carefully imitating a signature would in general be slower than the owner of the signature.

Another problem with resampling is that the critical points, capturing the characteristics of the signature, may be lost; critical points are sometimes added separately to the set of equidistant points obtained after resampling to solve this problem [13]. For instance Ohishi et al. don't do uniform resampling but resample data according to the curvature change between consecutive sample points [23].

2.4.2 Normalization

In systems where the user may have to sign on tablets with different active areas, signature size normalization may be required. People usually scale their signatures to fit the area available for the signature. However, size difference may be a problem in comparing two signatures. Generally, signatures are normalized with respect to both width and height, but scaling doesn't always solve the problem since the signature may have a different aspect ratio. Alternatively, signature size can be normalized according to one of the dimensions (width or height), which doesn't completely remove size characteristic of a writer. It is also known that, people doesn't equally scale their signatures with respect to width and height [7]. The signature size is considered to be a writer specific characteristics, i.e. writer may always sign only in large or small signatures, which should be preserved if there is no difference the sizes of the active areas of tablets, used in the system.

Normalization with respect to skew is a preprocessing technique commonly used for handwriting recognition. In handwriting recognition systems, this type of normalization is performed to recognize words independent of the writing style. However, skew normalization is not useful technique for signature verification, since the skew is a writer specific characteristic.

Size normalization is not performed in our system, since there is a consistency between the tablets we use.

2.4.3 Smoothing

Tablets which have low resolution may suffer from discretization errors, resulting in jagged signature trajectories. Extracting local features from jagged signature trajectories, and then using them for verification may lead to poor system performance. Hence, smoothing is required for low resolution tablets. Herbst et al. used cubic smoothing splines [11] to both interpolate signature data between discrete tablet grid points and smooth the data.

Jain et al. [13] has used a Gaussian filter to smooth the signature. Gaussian filter smooths out small fluctuations in the signal, while preserving its' overall structure. The x- and the y-direction of the signature were smoothed separately.

2.5 Feature Extraction

Feature extraction phase is one of the crucial phases of an on-line signature verification system. The discriminative power of the features and their resilience to the variation within the reference signatures of a writer, play one of the major roles in the whole verification process. While features related to the signature shape are not dependent on the data acquisition device, presence of dynamic features, such as pressure at the pen-tip or pen-tilt, depends on the hardware used.

As mentioned previously, features may be classified as global or local, where global features identify signature's properties as a whole and local ones correspond to a properties specific to a sampling point. As an example, signature bounding box, trajectory length or average signing speed are global features, and distance or curvature change between consecutive points on the signature trajectory are local features. Features may also be classified as spatial (related to the shape) or temporal (related to the dynamics).

More than 40 different features have been reported and used for on-line signature verification. Some of the earlier researchers have compared these features and proposed a sets of features most reliable for the verification [5, 13, 26]. Dolfing et al. used linear discriminant analysis to identify most discriminative features; Jain et al. and Rigoll et al. identified feature sets by evaluating their effect on verification performance of the proposed systems. Yang et al. have used Genetic Algorithm to

find the most useful features for on-line signature verification [36]. However, there is no publicly available on-line signature database and there are no standards on how skilled forgeries must be obtained, so it becomes difficult to justify which features are really discriminative and most suitable for on-line signature verification.

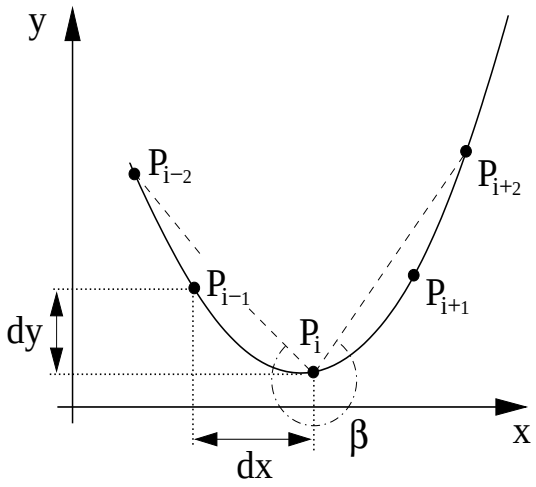

Figure 2.6: Local features extracted from an on-line signature trajectory.

Extracting and using only global features for verification is relatively easier and requires less computational resources than using of local features. However, global features alone lack discriminative power. We didn't use any global features in our method, all the features we have experimented with were local.

In our system we have experimented with the following local features of the sample points on the signature trajectory:

- x and y offsets relative to the first point on the signature trajectory
- x and y coordinate differences between two consecutive points
- curvature differences between two consecutive points
- critical points of signature trajectory

Figure 2.6 illustrates the curvature (β) and the differences in x,y coordinates (d_x, d_y) for the point P_i. P_{i-2} through P_{i+2} represent consecutive signature trajectory points. Each point has x and y coordinates and a time stamp as its initial features captured by the hardware. Time stamp represents a time instant when a point was recorded during the signature acquisition process. The curvature around the point P_i is obtained as the angle between the dashed lines.

All of the above mentioned features except critical points are calculated for each sample point on a signature trajectory. Critical points are a set of the sampling points which define signature's overall shape, and are described in Subsection 2.5.1. Feature vectors of each feature type are separately extracted and then used for calculation of the dissimilarity value between two signatures. Since signature data is not resampled in the system, feature vectors of length equal to a number of sampling points in a signature trajectory are extracted. Dissimilarity value calculation is described in Section 2.6.

2.5.1 Critical Points

Although different heuristics may be established to identify critical points of a signature trajectory, we prefer to call sampling points of high curvature as the critical points. Critical points, defined in this way, indicate crucial sampling points which determine overall shape (skeleton) of a signature. Rest of the points, which are around critical points, refine some subtle details of the signature shape. However, these non-critical points determine temporal features of a writer behavior, such as velocity or pressure change.

To identify critical points, all redundant points are first discarded from a signature trajectory. Redundant points are those consecutive points which indicate same coordinates of a signature trajectory but captured at different time periods. Redundant points are caused by slow signing speed of a writer and high sampling rate of the data acquisition hardware. Then the curvature is calculated for each remaining trajectory point. Finally, if the curvature difference between two consecutive points is higher than some threshold that point is identified as a critical, otherwise discarded. Figure 2.7 indicates critical points of the signature depicted on Figure 2.2.

2.6 Signature Dissimilarity Calculation

Now that the signature can be represented by the feature vector, we need a method to compare two signatures based on their vector representations. Aa was mentioned before, there is a variation among genuine signatures of a writer, which may result

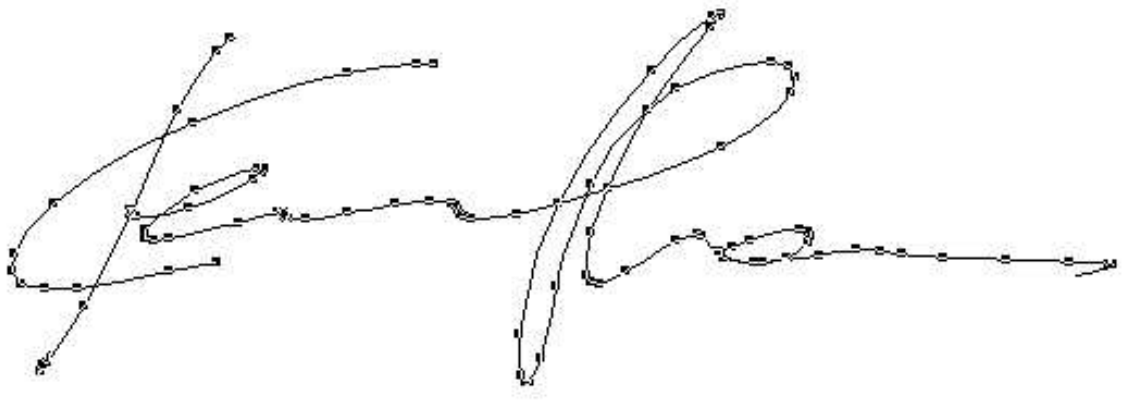

Figure 2.7: Critical points identified on an on-line signature trajectory.

in signatures of different lengths implying feature vectors of different lengths. Difference between the vector lengths makes it impossible to use a naive element-wise comparison of signatures, such as Euclidian distance between two vectors.

Taking into consideration that signature trajectory can be viewed as a function of time, a reliable dissimilarity comparison method must meet following criteria: (i) cross-over alignments between the points are not allowed, (ii) variation within genuine signatures must be taken in to consideration.

In our system, in order to compare two signatures, *Dynamic Time Warping* algorithm is used. This algorithm is a well-known and widely used method for aligning vectors of different lengths. Dynamic time warping algorithm finds the best non-linear alignment of two vectors such that the overall distance between corresponding vector elements is minimized in least square sense. The overall distance between two signatures S_1 and S_2 is calculated in linear time as shown in the following equation:

$$C[i,j] = Min \begin{cases} C[i-1,j] & + GapCost, \\ C[i,j-1] & + GapCost, \\ C[i-1,j-1] & + Dist(S_1[i], S_2[j]) \end{cases} \tag{2.1}$$

where $S_i[j]$ denotes the i'th signature's j'th point in the trajectory and

$$Dist(x,y) = \begin{cases} 0 & \text{if } ||x-y|| < Thr \\ ||x-y|| - Thr & \text{otherwise} \end{cases} \tag{2.2}$$

which is designed in a way to allow small variations between aligned elements.

The above formulae show the well-known dynamic programming algorithm, where C is the matrix to be filled by the algorithm and the $GapCost$ is the constant coefficient penalizing a missing or extraneous point in either of the signatures. The $Dist$ function is designed to allow for insignificant variation in reference signatures by using the constant Thr, which is the threshold defining maximum allowed dissimilarity between the aligned sampling points. The result of applying the dynamic programming algorithm ($C[length(S_1), length(S_2)]$) gives the dissimilarity score of two signatures, which we call a distance between two signatures.

One of the crucial points in the dynamic programming is the selection of gap penalties. Proper selection of gap penalties will enable the control over the alignment score, such that the score between forgery and genuine signatures will be high, whereas it remains low for genuine signatures. Different strategies can be followed in gap penalty selection:

- Constant gap penalty regardless of gap length.
- Larger gap opening penalty followed by a much smaller gap extension penalty.
- Gap penalty increasing rapidly with gap length.
- Different gap penalties for reference and test signatures.

The alignment score will linearly increase with respect to the gap length, using constant gap penalty. Using larger gap opening penalty followed by a much smaller gap extension penalty allows better control over a number of gap segments in the alignment. Gap penalty which is rapidly (exponentially) increasing with a gap length can maintain an alignment scheme where just small gap segments will be permitted. Using different gap penalties for a reference and a test signatures will give a control over a significance of a gap opening in either of the signatures.

2.7 Enrollment

During enrollment to the system, a user supplies a number of reference signatures, which are then used for calculations of user specific statistics describing the variation within reference signatures. Feature vectors are extracted from the reference

signatures, and are pairwise aligned using the dynamic programming algorithm. So if the user supplies N reference signatures, pairwise alignments of them result in $N(N-1)/2$ distances using which following statistics are being calculated:

- Average of distances to nearest signatures
- Average of distances to farthest signatures
- Average value of all pairwise alignments

The average of distances to nearest signatures is being calculated by averaging the distances from each reference signature to it's closest neighbor. The average of distances to farthest signatures is calculated in the same manner, however in this case distances to the farthest neighbors are being averaged. In addition to the averages, template signature is also selected amongst the reference set. The template signature is referred to be the one from which average distance to all other reference signatures is the minimum.

Figure 2.8 depicts the reference signatures supplied by a particular user, where x_i represents i'th reference signature supplied by the user, X_t represents the template signature. Min and Max represent maximum and minimum distances among the reference signatures, respectively. Dashed line with an arrow points to a nearest neighbor of a signature it is originating from. Average of distances represented by dashed lines with arrows gives the average of distances to nearest signatures.

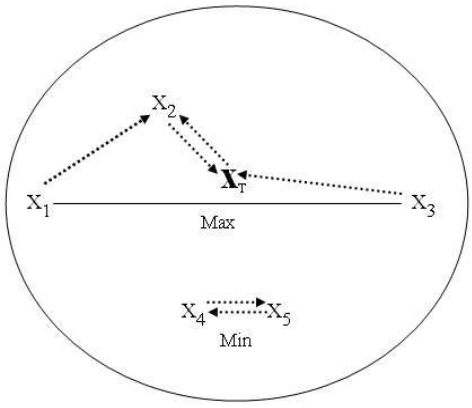

Figure 2.8: Distances between reference signatures used for user profile creation.

We store the reference signatures together with the mentioned average distances in a user profile, to later be used during the verification process. In return to the

signatures, the user receives an ID which defines his identity in the system.

2.8 Verification

During the verification phase, a test signature and an ID of a claimed user are submitted to the system. The test signature is compared with each reference signature, resulting in a number of distances. Out of these distances, distance to the closest, farthest, and the template signatures are all used to classify the test signature as genuine or forgery. Figure 2.9 depicts the distances used for the verification.

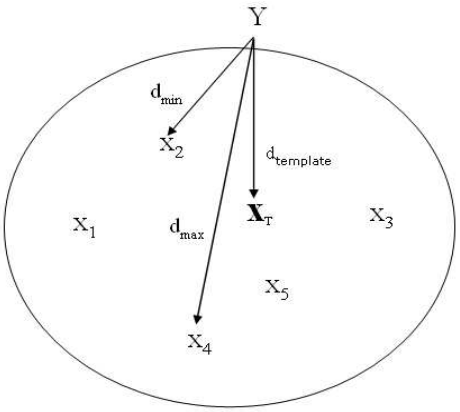

Figure 2.9: The distances used in the verification process. x_i represents i'th reference signature. Y and X_t denote the test and the template signatures, respectively. d_{max}, d_{min}, $d_{template}$ represent distances to the furthest, nearest and template reference signatures, respectively.

In previous systems, only one of these distances, typically the distance to the nearest reference signature or the distance to a template signature, was chosen in an ad-hoc manner, to classify the signature as genuine or forgery. In our system, we utilize all of these distances, treating them as features in a two-class classification problem, using standard pattern classification techniques, such as Bayes classifier, Support Vector Machines, and Linear classifier.

Since the dissimilarity score between two signatures directly depends on lengths of these signatures, the score must be transformed to a more robust feature. After studying and experimenting we came up with two transformations. In the first one, the distances are normalized by their corresponding averages in the reference set.

In the second one, the distances are transformed to their z-score, described in detail in 2.8.4. Applying these transformations eliminates the need for user-dependent thresholds.

The distribution of the normalized validation data, shown in Figure 2.10, supports that genuine and forgery signatures are well separated with the normalized features. Supplementary data distributions are given in appendix A. We have experimented with Bayes classifier, SVM (Support Vector Machine), and Linear classifier to robustly separate genuine and forgery distributions. In order to train and optimize the classifiers, a validation data set of 76 genuine and 54 forgery signatures, was constructed. Each validation signature was compared to the reference set of signatures of a user that the validation signature claimed to belong. The resulting distances of each validation signature were either normalized or transformed to z-scores, which then were used for training. Mentioned classifiers and the way they were trained are described in subsections 2.8.2, 2.8.3, and 2.8.1, respectively.

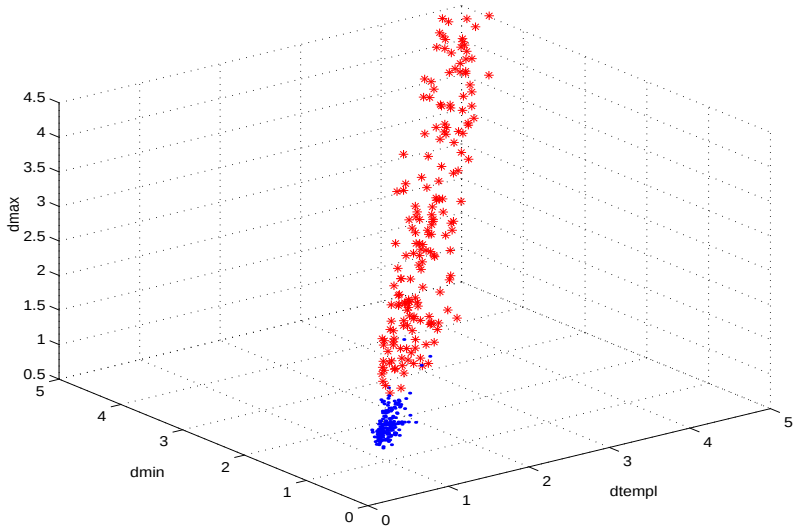

Figure 2.10: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional normalized distance vector, where dmax, dmin, and dtempl represent dimensions spanned by the corresponding normalized distances.

2.8.1 Linear Classifier

Principal Component Analysis or *Karhunen-Loeve Transformation* is a well known linear dimensionality reduction technique. Using PCA, high-dimensional data can

be projected on to a lower dimensional space which best represents the data in a least-squares sense. In other words, PCA, instead of selecting a subset of a given set of features, linearly combines them to produce a smaller set of features. Since theoretical background of PCA is beyond the scope of this work we will only explain practical application of the technique to the problem of signature verification. For details on PCA refer to [14].

The principal component was calculated using the feature vectors of signatures in the validation data set, where each vector was the set of the transformed distances. Same vectors were then projected onto the principal component and reduced to a one dimensional data. A threshold value, which well separated the genuine and forgery signatures, was empirically selected by examining the distributions of the projected vectors. The same threshold was then used in classifying the test signatures projected onto the same principal component.

As the three features were highly correlated (Figure 2.10), we could reduce the dimensionality from three to one while keeping most of the variance. This technique was separately applied to vectors of normalized distances and z-scores, respectively. The results are reported in section 2.9.

2.8.2 Bayes Classifier

Bayesian decision theory is a well known statistical technique used for pattern classification. Bayesian theory is deeply studied in [6], here we will only briefly provide with the necessary terminology and describe the way the Bayes classifier is used for on-line signature verification problem.

A *prior probability* of a signature class (genuine or forgery) represents the frequency of that particular class being selected each time a signature is verified. In our case, each class has equal prior probability, since each time a signature is verified there is an equal chance of classifying that signature as genuine or forgery. C_f and C_g denote class labels of forgery and genuine signatures, respectively and $P(C_k)$ denotes prior probability of a class k (genuine or forgery).

A probability that a signature has a feature vector X and belongs to a class k is called a *joint probability* and is denoted by $P(C_k, X)$. A a probability that a signature has a feature vector X given that it belongs to a class k is a *conditional*

probability and is denoted by $P(X|C_k)$. The difference between joint and conditional probabilities is that the joint probability is calculated over all signatures, while conditional probability is calculated only over a particular class of signatures. Besides, the joint probability can be expressed in terms of the conditional and the prior probabilities in the following way:

$$P(C_k, X) = P(X|C_k)P(C_k) \tag{2.3}$$

or

$$P(C_k, X) = P(C_k|X)P(X) \tag{2.4}$$

where $P(C_k|X)$ is called a *posterior probability*, and $P(X)$ is a probability of observing feature vector X over all signatures (genuine and forgery). The posterior probability represents a probability that a signature belongs to a class k given that a feature vector X was measured for it. Using the above two equations, posterior probability can be expressed as:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{2.5}$$

The Bayes theorem states that classifying a signature to a class having the largest posterior probability minimizes the misclassification error, i.e. decide C_g if $P(C_g|X) > P(C_f|X)$ otherwise decide C_f. In addition to that, posterior probabilities can be expressed in terms of probabilities which are relatively easier to calculate.

We have modeled posterior probabilities of genuine and forgery signature distributions using *Gaussian* densities. A d-dimensional Gaussian density, namely a *Multivariate Normal Distribution*, is represented in Equation 2.6, where x is a d-dimensional feature vector (3 in our case), Σ denotes covariance matrix, μ is a mean vector, $|\Sigma|$ and Σ^{-1} represent determinant and inverse of the covariance matrix, respectively.

$$p(x) = \frac{1}{2\pi^{d/2}|\Sigma|^{1/2}} \exp[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)] \tag{2.6}$$

The parameters of the Gaussians were estimated using validation data set. Generally, to ease the parameter estimation phase natural logarithm of the Gaussian is used:

$$g_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_k| + \ln P(C_k) \tag{2.7}$$

Since the prior probabilities of genuine and forgery signatures are assumed to be equal, second and forth terms of the above equation can be removed. The discriminant function is defined as $g(x) = g_g(x) - g_f(x)$, where test signature will be classified as genuine if $g(x) > 0$ or forgery otherwise. Classification results using Bayes classifier are reported in section 2.9.

2.8.3 Support Vector Machine

Support Vector Machine (SVM) is a relatively new pattern classification technique, based on statistical learning theory [4, 31, 34]. SVMs are applicable to regression and classification tasks where they have consistently shown higher performance than traditional learning tools (especially for classification problems).

The basic idea of SVMs is that, SVM maps the input space into a higher dimensional feature space. Mapping can be done either linearly or non-linearly, according to the kernel function used for the mapping. In this new feature space, the SVM constructs separating hyperplanes that are optimal in the sense that the classes are separated with the largest margin and minimum classification error. The optimal hyper plane can be written as a combination of a few feature points those are called the *support vectors* of the optimal hyper plane.

In our system SVM was trained using the 3-dimensional feature vectors of the transformed distances which were calculated using signatures of the validation data set. Later, same SVM was used to classify signatures of the test data set.

2.8.4 Z-Scores

Standardized score or *z-score* is known to represent relative status of that score in a distribution. In our case, z-score indicates a deviation of a distance from a mean of its distribution in a standard deviation units. In other words, z-score indicates how many standard deviations away is a distance from its mean. Z-Score is calculated as fallows:

$$Z = \frac{x - \mu}{\sigma} \tag{2.8}$$

Converting the distances to their z-scores will transform the distribution of genuine signatures of a particular writer to a distribution of 0 mean and a standard

deviation of 1. This transformation doesn't alter the form of the original distribution, since the frequency of any given z-score is equal to that of the distance it corresponds to in the original distribution.

2.9 Performance Evaluation

Here we describe a strategy our system's performance is evaluated. Data sets used for the evaluation and performance results are described in subsections 2.9.1 and 2.9.2, respectively.

In evaluating the performance of a signature verification system, there are two important factors: the false rejection rate (FRR) of genuine signatures and the false acceptance rate FAR) of forgery signatures. As these two are inversely related, lowering one often results in increasing the other. Hence, it is common to talk about the equal error rate which is the point where FAR equals FRR. Since obtaining actual forgeries is difficult, two other forgery types have been defined and used in performance evaluations. A *skilled* forgery is signed by a person who has had access to a genuine signature for practice. A *random* or *zero-effort forgery* is signed without having any information about the signature, or even the name of the person whose signature is forged. While any signature in a database can be used as a random forgery to everyone besides its owner, obtaining truly skilled forgeries is difficult, since subjects forging signatures to help build a signature database are not expected to be as ambitious as an actual forger. Therefore, it may be more suitable to talk about random and *informed* forgeries.

2.9.1 Data Sets

The system performance was evaluated using sample signatures supplied by 94 subjects enrolled in our system. 73 of the enrolled subjects are students (graduate and undergraduates), faculty members, and workers of Sabanci University, where the remaining 21 were enrolled to the system during the demonstration of the system in CeBIT 2002, in Istanbul. Figure 2.11 depicts sample signatures of some users who contributed to our signature database.

There were no constraints on how to sign and no information about the working

mechanism of the system were supplied to the subjects, so that they signed in their most natural way. Each subject supplied 10 to 15 genuine signatures 8 of which were used for profile creation and the rest for the performance evaluation of the system, which comprise our first data set (G1) of 182 test genuine signatures.

To collect skilled forgeries we added a signing simulation module to our system. Simulation module animates the signing process of a signature so that the forger is able to see not only the signature shape but the signing dynamics (speed and acceleration) as well. Signature simulation module animates signature dynamics according to the time stamps of signature's trajectory points. Forgers had a chance to see the animation several times and practice tracing over the signature image a few times before forging it. Our forgery data set (F1) consists of 313 skilled forgeries obtained in this way.

A data set (G2) of 124 genuine signatures, which were collected from the subjects who enrolled to the system more than six months before their contribution to the G2 (3-5 signatures from each). The aim of constructing this data set was to test the system against possible changes in signatures over time.

A validation data set of 76 and 54 genuine and forgery signatures, respectively was constructed to train some of the classifiers and estimate their parameters.

Finally, to evaluate the system performance against random forgeries we used each genuine signature in the database as a forgery for users beside the owner. These random forgeries may be considered as a data set (F2) of totally 69936 signatures. All data sets mentioned above are summarized in Table 2.2. Note that, even though this is not a very large set, there is no public online signature database (that we know of).

Data Set	*Signature #*	*Type*
G1	182	Genuine
F1	313	Skilled forgeries
G2	124	Genuine
F2	69936	Random forgeries

Table 2.2: Data sets used to evaluate on-line signature verification system's performance.

Figure 2.11: Sample signatures of some users who contributed to the signature database.

2.9.2 Results

Results reported here are based on the verification of test signatures, where each signature was represented by a feature vector of elements corresponding to x and y coordinate differences between two consecutive trajectory points. Performance results of the system based on the other features, which were described in Section 2.5, are reported in Appendix B.

Table 2.3 summarizes performance results, where the best results were obtained using the Linear classifier. Overall performance error rate for the Linear classifier is 1.46%, which is a good result compared to the state-of-the-art performance results.

First three rows of the Table 2.3 are the results of the classifiers using the 3-dimensional feature vector of the normalized distances, and the following two rows

using the feature vector of the distances converted to the z-scores. Results on z-scores are inferior to the normalized distances, which is due to z-score's property of directly incorporating variance of a distribution. Hence, the signatures of people having large variation within their reference signatures can be forged more easily if z-scores are used. Since our data sets contain genuine and forgery signatures separately, only false accept error rate (FAR) or false reject error rate (FRR) is reported for each data set. Overall error rate represents system's error calculated by averaging classification errors in G1, F1, and G2 data sets.

Classifier	*G1: FRR*	*F1: FAR*	*G2: FRR*	*Overall Error Rate*
Linear	1.65%	1.28%	1.61%	1.46%
Bayes	2.19%	3.52%	5.64%	3.60%
SVM	0.55%	3.85%	3.33%	2.75%
PCA on Z-Scores	2.88%	4.39%	8.87%	4.85%
Bayes on Z-Scores	4.39%	6.73%	6.45%	5.99%

Table 2.3: System performance results using the classifiers mentioned in section 2.8 and d_x, d_y in feature vectors.

The system's performance was also evaluated against random forgeries using the data set F2 and the Linear. False accept error rate was measured to be 1.06%. This result seems to be unexpected when compared with the results on data set F1, where forgeries were considered to be skilled. However, after the investigation of the error cause, we found that only a few users who supplied very inconsistent reference signatures have contributed to the reported false accept error rate.

The relative performances of the different classifiers are not very significant since the database is rather small and the results are very close; rather, the main goal of this work was the design of a dissimilarity measure that separates genuine and forgery classes quite well.

2.10 Summary

A system with improved decision criterion and classification methodology relying solely on standard pattern recognition techniques is implemented for on-line signature verification.

Two different pressure sensitive tablets are used as the input devices to the system. Each tablet is able to sample 100 sampling points per second, where each sampling point has x,y coordinates and a time stamp as features. Different preprocessing techniques commonly used in previous works are discussed. In our system, we found that the advantage of not resampling, namely keeping all the dynamic information, significantly outweighs the advantages of resampling. Besides, difference in coordinates between two consecutive trajectory points were found to be useful for signature representation, as they are more robust to local variations of signatures. The dissimilarity between two signatures is established using dynamic time warping algorithm, which finds the best non-linear alignment of two vectors of different lengths such that the overall distance between corresponding vector elements is minimized in least squares sense.

During the verification, a test signature and an ID of a claimed user is being submitted to the system. After the alignment of the test signature with the reference set signatures, distances to the nearest, farthest, and template signatures are recorded. In previous systems, only one of these distances, typically the distance to the nearest reference signature or the distance to a template signature, was chosen, in an ad-hoc manner, to classify the signature as genuine or forgery. In our system, we utilize all of the distances using standard pattern classification techniques. The distances are first normalized by the corresponding reference set averages, resulting in a three dimensional space where genuine and forgery signature distributions are well separated.

We have experimented with Bayes classifier, SVM, and Linear classifier to verify the test signature. To report unbiased system performance results, all of the classifiers were trained using validation dataset and tested on separate test data sets. We have collected 619 test signatures (genuine and forgery together) from 94 enrolled users. Since it is very difficult to obtain real forgeries, we obtained skilled forgeries which are supplied by forgers who had access to signature data to practice

before forging. To collect forgery signatures a special module is added to the system, which is animating presented signature according to time stamps of sampling points. Forger before submitting a signature had a chance to see the animation for several times. Best results of 1.46% EER are obtained with the Linear classifier. The results show significant improvement over the results of reported state-of-the-art systems.

Chapter 3

Off-Line Signature Verification

This chapter describes our system proposed for an off-line signature verification. In Section 3.1 we make a literature overview existing methods for the off-line signature verification. In Section 3.2, brief description of the system and its main modules are presented. Section 3.3 describes image preprocessing steps required to prepare signatures for the feature extraction phase. Feature extraction phase is described in Section 3.4, followed by the method designed to establish similarity degree between two signatures, covered in Section 3.5. Enrollment to the system and the verification processes are discussed in sections 3.6 and 3.7, respectively. The performance results of the system are reported in Section 3.8.

3.1 Literature Overview

Automatic off-line signature verification is a very old pattern classification problem, involving the discrimination of genuine and forgery signatures, written on a piece of paper. Unlike on-line systems, off-line systems have only the image of a signature as input; in other words, dynamic information is not available for the off-line signature verification. Other difficulties such as variation within genuine signatures, noise introduced by the scanning device, or a difference in pen width make off-line signature verification a challenging problem. It is worth to notice that, even professional forensic examiners perform at about 70% of correct classification rate (genuine or forgery). The difficulty of the classification can be appreciated by looking at the Figure 3.1, which depicts four genuine and a test signatures. Although the test signature seems to be authentic, it is actually a forgery.

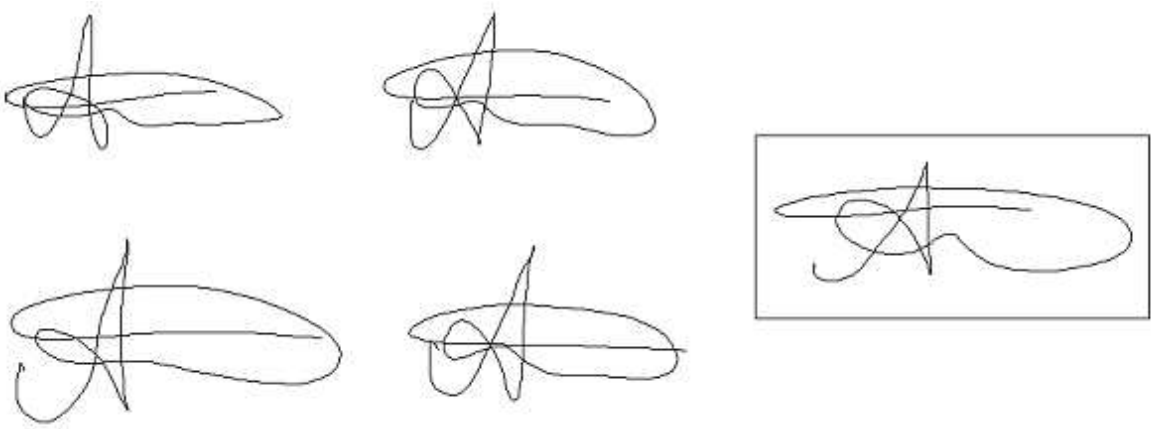

Figure 3.1: The figure depicts the difficulty of the signature classification. The variation in the four reference signatures (at left) makes it difficult to classify the test signature (at right).

Due to the difficulties of the problem, much of the research effort in the area has been spent on random and simple (unprofessional) forgery detection, where deceit is generally obvious. Unlike random forgery detection, skilled forgery detection is a much more difficult task. Another interesting fact about signatures is that no two genuine signatures are exactly the same; when two identical signatures found, forensic examiners consider one as a trace over the other. Thus, a signature verification system should be able to differentiate between the variation within genuine signatures and the fraud. Off-line signature verification involved in skilled forgery detection is still an open research question. Following is a brief review of the methods proposed for random and skilled forgery detections, respectively.

3.1.1 Random Forgery Detection

Since random forgeries differ significantly from the genuine signatures, and to a lesser extent simple forgeries, some of the earliest systems used only image based features for off-line signature verification. For instance, Revillet [25] used moment invariants and Fourier descriptors to classify signatures as genuine or forgery. Nemcek [22] used a maximum likelihood classifier, applied onto a number of features extracted from the Hadamard Spectrum, to detect simple forgeries. Performance of Nemcek's system had a false reject rate of 11% and false accept rate of 41%. Chuang [3] divided a signature into three regions (upper, middle, and lower) out of which a number of

global features were extracted. The features and their ratios were compared with those of the reference signature, using weighted distance as the dissimilarity metric. Chuang reports a 20% equal error rate for the proposed method. As can be inferred from the results of the systems, using only global image features is not sufficient even for random forgery detection.

Most of the remarkable work in the area of random forgery detection was done by Sabourin et al. In [27], Sabourin et al. used extended shadow codes, previously proposed by Burr for handwritten character recognition [2], as a shape feature to detect random forgeries. These codes incorporate both global and local representation of a signature. To calculate shadow codes the image is projected onto a bar mask array, where each bar is associated with spatially constrained area of a signature. Shadow projection was defined as the simultaneous projection of each signature pixel onto it's closest horizontal, vertical, and diagonal bars. Feature vector of normalized shadow codes is then used by a k-Nearest Neighborhood and a minimum distance classifiers. Best system performance results of 2.16% equal error rate were obtained using k-NN classifier. However, extended shadow codes are poorly tolerable to changes in signature translation, rotation, and scale.

Sabourin et al. also introduced an identity grid, separately for each writer [20]. Identity grid divides a signature in to a number of cells such that grid reflects overall shape of the signature. Cells are then presented to a neural network, which reduces cell sizes by 1/3. After the size reduction, for each cell a Fuzzy ARTMAP network is trained to learn its contents. During the verification, outputs of each ARTMAP are combined to produce the final decision. System's performance results are reported to be 9.14% of equal error rate.

Most of successful works of Sabourin et al. were based on shape matrices [30] and a local granulometric size distributions [28,29]. Shape matrices were previously used for planar shape representation of industrial parts and machine printed characters. In the proposed method, shape matrices were used as a mixed shape feature for signature verification. Mixed shape feature is actually a global feature, in calculation of which positions of local features are taken into account. Best results of 0.84% equal error rate were reported using a test data set of 800 signatures. In the second work, a signature was centered on to a grid of rectangular retinas, which were excited

by the signature's trajectory pixels at that location. Granulometric size distribution were used as the shape descriptors of each retina, where granulometry is the result of a set of morphological openings. Finally, a feature vector of dimension equal to a number of retinas was used by k-NN classifier to detect random forgeries. The system's performance was evaluated using the same test data set and the best result reported to be a 0.02% of equal error rate. Although very good results were obtained from both systems, they both assume that all signatures are of similar size, and that the corresponding strokes of the signatures fall into approximately the same retinas.

Other methods which mainly aimed skilled forgery detection, and also reported their results on random forgery detection, will be discussed in subsection 3.1.2.

3.1.2 Skilled Forgery Detection

Signatures in off-line systems usually may have noise, due to scanning hardware or paper background, and contain less discriminative information since only the image of the signature is the input to the system. While genuine signatures of the same person may slightly vary, the differences between a forgery and a genuine signatures may be imperceptible, which make automatic off-line signature verification be a very challenging pattern recognition problem. Besides, the difference in pen widths and unpredictable change in signature's aspect ratio are other difficulties of the problem.

Herbst et al. proposed to use the Radon transform and the dynamic programming algorithm to detect skilled forgeries [12]. A sinogram was constructed by applying the Radon transform to a signature image, where the signature can be reconstructed from it's sinogram by applying inverse Radon transform. In order to compare two signatures, their corresponding projections (rows of their sinograms) were used as input to the dynamic programming algorithm. The dissimilarity of two signatures was obtained by averaging alignment scores of corresponding projections. The system was tested using a data set of 460 genuine, 138 skilled forgery, and 138 random forgery signatures. Herbst et al. reported a 23% equal error rate when only skilled forgeries were used, and a 10% equal error rate when only random forgeries were used.

Guo et al. [10] approached the problem by establishing a local correspondence between a model and a questioned signature. The questioned signature was seg-

mented into consecutive stroke segments that were matched to the stroke segments of the model. Stroke segment boundaries were defined by topological features, such as crossings and endings. The cost of the match was determined by comparing a set of geometric properties of the corresponding sub-strokes and computing a weighted sum of the property value differences. The least invariant features of the least invariant sub-strokes were given the biggest weight, thus emphasizing features that were highly writer-dependent. Random and simple forgeries were detected when a good correspondence couldn't be found. The threshold between genuine and forgery signatures based on the correspondence cost was determined by a Gaussian statistical model. Skilled forgery detection was performed by examining the writer-dependent information embedded at the sub-stroke level and trying to capture unballistic motion and tremor information in each stroke segment. Guo et al. reported a 0.13% false accept rate and 0.39% false reject rate using Sabourin's data set of 800 test signatures (genuine and random forgeries). For the case of skilled forgeries a 6% false accept rate and 11% false reject rate was reported using a data set of 50 skilled forgery and 200 genuine signatures.

Fang et al. [8] used vertical and horizontal projections of signatures for skilled forgery detection. To compare two signatures their corresponding projections were aligned using dynamic programming algorithm. Wrapping function of the test signature was compared to the average wrapping function of reference set signatures using the Mahalanobis distance. A data set of 1320 genuine and 1320 forgery signatures was used to test the system. Fang et al. reported the best results of 23.2% false reject rate and 21.4% false accept rate, obtained using both vertical and horizontal projections

Mizukami et al. [18] used displacement extraction method to establish dissimilarity between two signatures. First, two dimensional displacement function between two signature images was found using Gauss-Seidel iterative approximation method. Displacement function matched corresponding coordinates between test and reference signatures. The dissimilarity between two signatures was calculated using sum of Euclidian distances between corresponding signature coordinates, which were prior adjusted using displacement function. A data set of 200 genuine and 200 skilled forgery signatures was constructed to test the system. Mizukami et al. re-

ported performance result of a 24.9% average error rate.

3.2 General System Overview

Off-line signature verification has a significant use mainly in establishing the authenticity of bank checks and other official documents, based on the signatures they carry. For instance, thousands of checks are being processed in a day in most banks or insurance companies; hence there is a great need for the automation of this process. In the proposed system, instead of classifying a signature only as genuine or forgery, the signature is being classified into three classes: genuine, forgery, and uncertain. In this way, signatures which can be identified as genuine and forgery with confidence will be separated, and a human examiner (ex. bank officer responsible from signature inspection) will examine only a small portion of signatures classified into the third class of uncertain signatures.

Figure 3.2 depicts a high level representation of the proposed off-line signature verification system. The system can be viewed in three modules: image acquisition and preprocessing module, enrollment and verification modules. Image acquisition and preprocessing module is responsible for extracting signature image from a scanned document and afterwards preprocessing it for farther feature extraction. Any ordinary scanner with enough resolution can be used as an image acquisition device. During preprocessing, noise possibly introduced by the scanning hardware is removed and the signature is binarized.

Before verification or enrollment can take place, features are needed to be extracted from the signature's image. In on-line signature verification, feature extraction was a relatively easy phase as we had signature trajectory points indexed in a vector according to their sampling time stamps. For the off-line case, we have only a signature image to work on, where it is very hard to robustly extract permanent and genuine signer's unique features. We have extracted and experimented with signature envelopes and various projections as the features. Feature extraction phase is covered in section 3.4.

During enrollment, user supplies a number of reference signatures which are used by a profile generator to create a profile on the system for that user. Profile contains

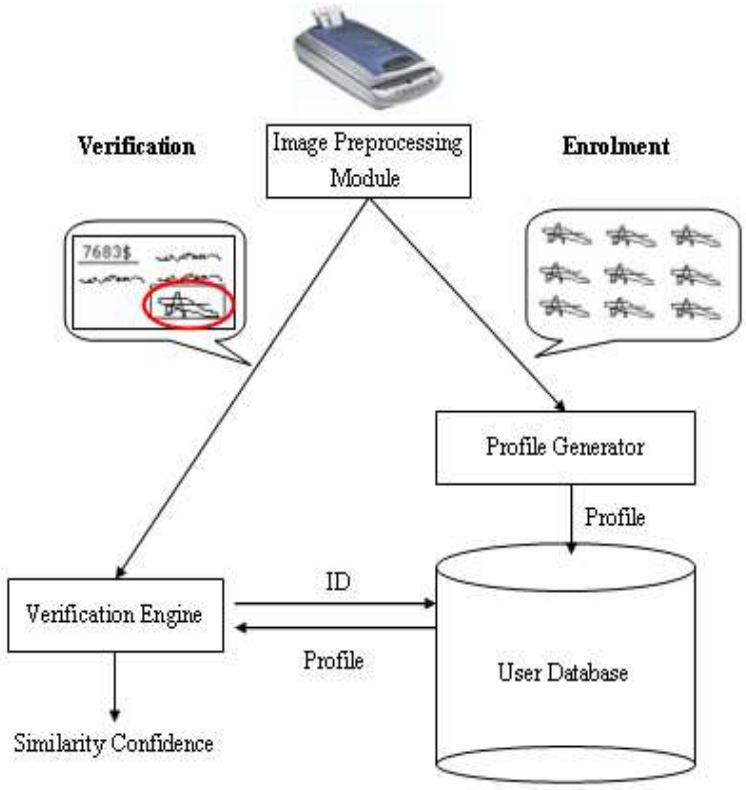

Figure 3.2: High level representation of the proposed off-line signature verification system.

the reference signatures and similarity values. Similarity between two signatures is calculated using the dynamic programming algorithm. Similarities among all reference signatures are further transformed to calculate similarity scores (distances) which describe variations within reference signatures. Dynamic programming algorithm is broadly described in section 2.6. More detailed information on similarity values and how they are being extracted is given in section 3.5. Enrollment phase to the system is covered in section 3.6.

Verification engine is used to authenticate a given (test) signature against the claimed ID. Firstly, reference set signatures and similarity values corresponding to the claimed ID, are retrieved from the system's database. Then, using dynamic programming algorithm, the test signature is compared with each reference set signature. Comparison results in a number of dissimilarity values (distances), which further normalized and presented to a classifier as a feature vector. We have experimented with a linear classifier used in conjunction with Principal Component Analysis, to classify a test signature as genuine or forgery. Normalization of the

similarity values, and the way they were used by the classifier are broadly described in section 3.7. Finally, performance evaluation and the data set used to evaluate system's performance are reported in section 3.8

3.3 Preprocessing

Any ordinary scanner with enough resolution can be used as an image acquisition device. Scanning hardware may introduce noise to a signature image. Another source of noise may be speckled paper background on which the signature is signed. Noise on a signature image may thwart feature extraction process; hence it needs to be removed. However preprocessing methods should be selected carefully as they may remove signature properties peculiar to a signer.

Although we don't know the real noise distribution, we have used a Gaussian filter to smooth the image of a signature. Gaussian smoothing filter is known to be a very successful in normally distributed noise removal. Two-dimensional, zero-mean Gaussian function is defined as:

$$g[i,j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \tag{3.1}$$

where σ is the Gaussian spread parameter, determining the width of the Gaussian. Since Gaussian function is symmetric, smoothing performed by the filter will be the same in all directions thus edges in an image will not be biased in some particular direction, which is important. After the smoothing, image is binarized by a simple thresholding scheme.

Even after smoothing some small dots and isolated pixels may remain due to binarization; hence we decided to use the morphological opening and closing operators to get rid of these spurious data. Image morphology is beyond the scope of this work, hence we will only briefly describe the open and close morphological operators to give some insight; for broad coverage on morphology refer to [9]. The opening operator with a given structuring element will remove all the points which are too small to contain that structuring element. The closing operator, in the contrary, fills in holes and concavities smaller than that structuring element. Both close and open operators smooth an image. Selection of structuring elements is a crucial step, where if wrongly selected may cause distortions and deteriorate system performance.

Signature's size is considered to be one of the writer specific characteristics. We preserved this characteristic in our on-line system by not doing size normalization, since both tablets we used had approximately the same area for signing. However in off-line signature verification systems, there is no reliable way to control change in signature's size, since signing area provided to gather reference signatures and test signatures will be different in real applications. Scale difference of reference set signatures and the test signature may severely affect feature extraction and the similarity comparison phases. In the proposed off-line signature verification system, signatures are normalized with respect to both width and height. We broadly discuss signature normalization process in section 2.4.2.

3.4 Feature Extraction

In this work four different features were extracted for signature verification: upper envelope, lower envelope, vertical, and horizontal projections. These features were commonly used for handwriting recognition.

Upper envelope is the curve connecting upper most pixels of the signature trajectory. Likewise, lower envelope is the curve connecting lower most pixels of the signature trajectory. Figure 3.3 depicts a sample off-line signature, following is the Figure 3.4 which is to depict the upper and lower envelopes extracted for that signature. To extract upper envelope, each column of the image is traversed from top to bottom. The location of first encountered non-white pixel is marked as a point of the upper envelope. In the same manner, to extract lower envelope of a signature, each column of the image is traversed from bottom to top, recording first encountered non-white pixels to the envelope curve. The last figure 3.5 depicts the horizontal and vertical projection profiles of the same sample signature. Projection profiles are very sensitive to a pen width; hence to overcome this problem both vertical and horizontal profiles were normalized to signature's height and width, respectively.

As was mentioned in the previous section, before features are actually extracted the image of a signature is normalized, thus feature vectors are of fixed length. After feature extraction, we merge these feature vectors into a single vector, to be later used in similarity comparisons.

Figure 3.3: Sample off-line signature.

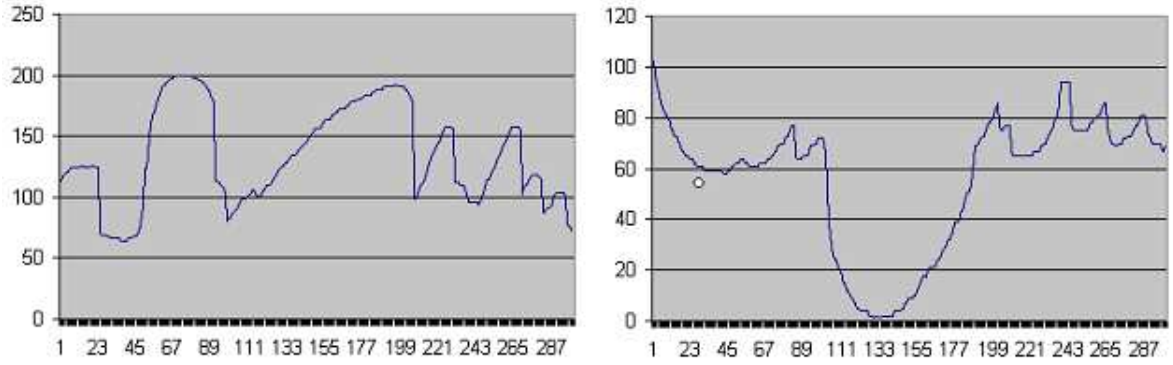

Figure 3.4: Upper and lower envelopes of the signature shown in Figure 3.3.

3.5 Signature Dissimilarity Calculation

Once the feature vectors are extracted, there is a need for a method two compare two signatures using their feature vectors. Since the feature vectors are of the same length, the most natural way to compare them would be to use the Euclidian distance between them. However this would lead to a very poor similarity metric since the corresponding features in the vectors may be distorted due to the natural

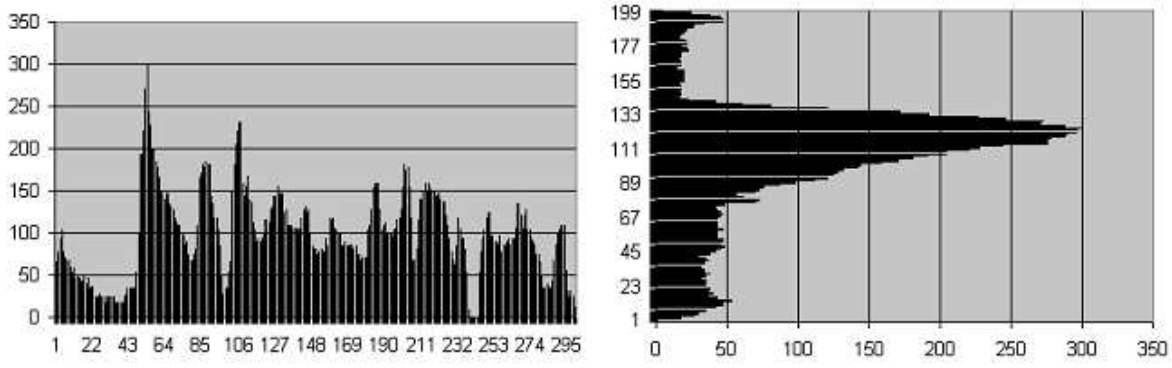

Figure 3.5: Horizontal and vertical projection profiles of the signature shown in Figure 3.3.

variance among genuine signatures. Second straight-forward way to compare the vectors would be to use autocorrelation. However autocorrelation misses the ability to properly deal with non-linear distortions in the signatures and has the principal area of application in recognition rather than verification, where in recognition the task is to find the closest template signal when an unknown signal is presented. Figure 3.6 depicts the lower envelopes extracted from two signatures which belong to the same person, and justifies that if Euclidian distance or autocorrelation would be used similarity score between these two feature vectors would be low.

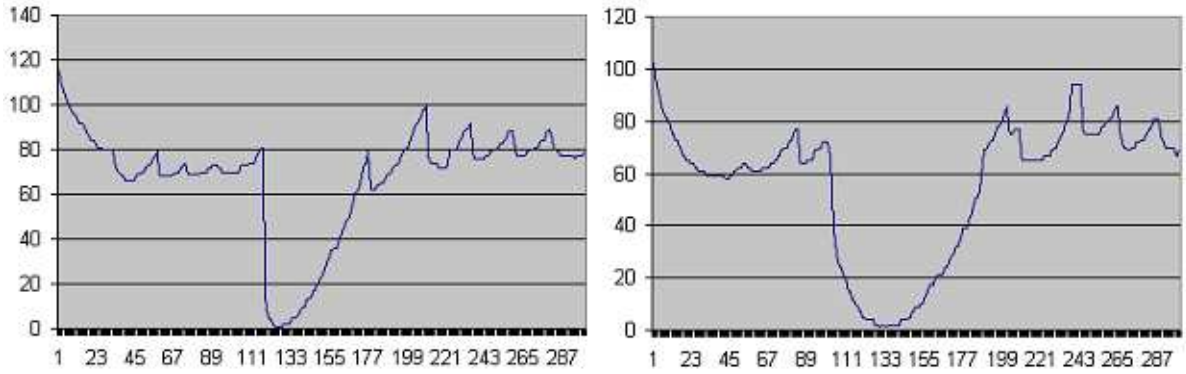

Figure 3.6: Two lower envelopes, corresponding to a two signatures of a same person, that would give a low similarity score if Euclidian distance or autocorrelation were used.

Hence, there is need for a method which can relate corresponding features of two vectors in a non-linear manner. In this system, we have used the Dynamic Time Warping algorithm, which finds the best non-linear alignment of two vectors such that the overall distance between corresponding vector elements is minimized in a least squares manner. Dynamic time warping algorithm was described in great detail in Section 2.6.

3.6 Enrollment

During enrollment to the system, a user supplies a number of reference signatures, which are then used for calculations of user specific statistics describing the variation within reference signatures. Feature vectors are extracted from the reference signatures, and are pairwise aligned using dynamic programming algorithm as described

in section 2.6. So if the user supplies N reference signatures, pairwise alignments of them result in $N(N-1)/2$ distances out of which following statistics are being calculated:

- Average of distances to nearest signatures
- Average of distances to farthest signatures
- Average value of all pairwise alignments

Average of distances to nearest signatures is being calculated by averaging the distances from each reference signature to it's closest neighbor. Average of distances to farthest signatures is calculated in the same manner, however in this case distances to the farthest neighbors are being averaged. In addition to the averages, template signature is also selected amongst the reference set. The template signature is referred to be the one from which average distance to all other reference signatures is the minimum.

After feature vectors are extracted, signature images are of no interest, since feature vectors are what is used in verification. We store the feature vectors together with the mentioned distances in a user profile, to later be used during the verification process. In return to supplied reference signatures, the user receives an ID which defines his identity in the system. For the case of bank checks, ID may be a customer number, which can be printed on to all checks before a check book is given to a customer.

3.7 Verification

Promising performance results were obtained from our on-line signature verification system. This inspired us to adopt the same strategy for off-line signature verification as well. After a signature is segmented from a document, preprocessed and the feature vector extracted out of it, system retrieves a profile corresponding to the claimed ID and the test vector is compared with each reference feature vector resulting in a number of distances. Out of these distances distance to the closest, farthest, and a template signatures are then used to classify the test signature. Normally it would be convenient to classify signature as genuine or forgery. However for off-line

signature verification it would be more useful if the system could robustly classify large portion of presented signatures as obviously genuine or forgery and leave the rest (hopefully a small portion) for further human inspection. Thus, signatures in our system are classified into three classes: genuine, forgery, and uncertain signatures.

Before distances are used in the classification, they are normalized by the corresponding averages of the reference set. Normalized distances are then put in to three dimensional feature vector, which is used by a classifier. The distribution of the normalized validation data, shown in Fig. 3.7, supports that obviously genuine and forgery samples are well separated with the normalized features and only a small portion of signatures is left for human check. Note that by normalizing the measured distance vectors by the corresponding reference set averages eliminates the need for user-dependent thresholds.

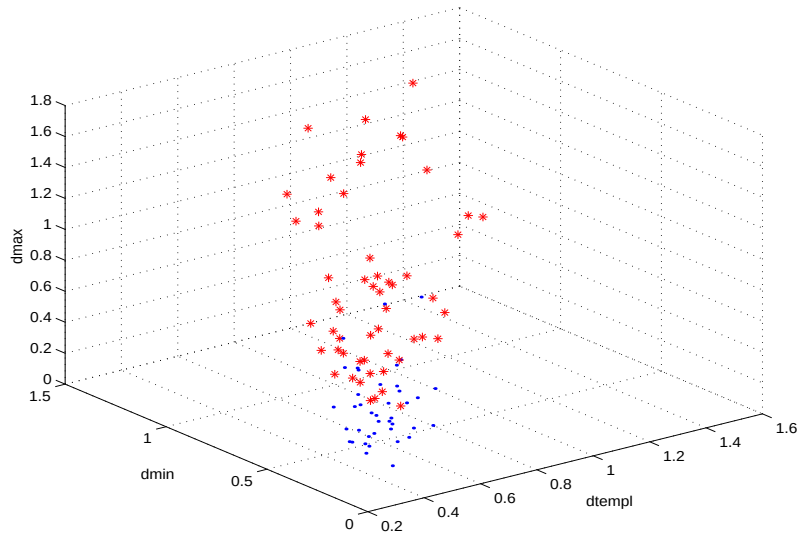

Figure 3.7: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional normalized distance vector, where dmax, dmin, and dtempl represent dimensions spanned by the corresponding normalized distances.

Firstly, to classify a signature, three dimensional feature vector is reduced to one dimension using PCA (Principal Component Analysis). PCA projects high-dimensional data on to a lower dimensional space, which best represents data in the least-squares sense, as described in more details in Section 2.8.1. To find the principal component and threshold values, validation data set consisting of 20 gen-

uine and 20 forgery signatures was used. Each validation signature was compared to the reference set of signatures it claimed to belong, giving a 3-dimensional feature vector (min, max, and template distance), which was normalized as mentioned above. Then normalized feature vectors were used to find the principal component and farther projected on to it resulting in one dimensional data. Finally, threshold values which define boundaries between genuine, forgery, and uncertain signature classes were empirically selected from projected one dimensional data. Same principal component and the thresholds calculated using validation data set were used to classify test signatures. Performance results are reported in section 3.8.

3.8 Performance Evaluation

In this section, we present performance results of the proposed system for off-line signature verification. In section 2.9 we have already mentioned about difficulties and strategy to evaluate performance of a biometric based security systems, in particular on-line signature verification systems. One of the main difficulties was to conduct a real field test, where skilled forgers occasionally would try to break into the system while genuine users routinely authenticate themselves.

3.8.1 Data Sets

We have constructed 3 data sets to evaluate the system performance, where 20 people contributed. Each person supplied 9 genuine signatures signed with 4 different pens. The reason behind using different pens was to test system's resilience against the changes in pen width. For each participant 6 out of the 9 signatures were randomly selected and used as a reference set. 2 other genuine signatures from each signer were set aside to construct first data set (G1) of 40 genuine signatures. Last genuine signatures of each user, constituting a set of 20 genuine signatures, were used for validation purposes. Each participant was asked to forge someone else's signatures, where two types of forgeries were collected: *skilled* and *random*. Forgeries supplied by a person who had a signature to practice before forging it are considered to be the skilled forgeries. In the case of random forgeries, people were only supplied with the name of a genuine signature owner, before they forged it. Defining forgeries in

this way, data sets of 60 skilled (F1) and 20 random (F2) forgeries were constructed. Also another set of 20 skilled forgeries were set aside for validation purposes. All the data sets we use, are disjoint. The table 3.1 summarizes data sets used to evaluate system performance.

Data Set	*Signature #*	*Type*
G1	40	Genuine
F1	60	Skilled Forgery
F2	20	Random Forgery

Table 3.1: Data sets used to evaluate off-line system performance.

3.8.2 Results

Normally to evaluate a biometric based security system two criteria are important: false accept rate and false reject rate. However for off-line signature verification systems, it would be more useful if the system could robustly classify a large portion of the presented signatures as obviously genuine or forgery and leave the rest (hopefully a small portion) for further human inspection, since even human examiners do that. In this way, the system could save much time for check examiners, working in banks and other companies where thousands of the checks must be processed each day.

Tables 3.2 and 3.3 summarize performance results of the system. Using all of the features together (envelopes and projections) we got a 15% error rate rejecting skilled forgeries; while non of the genuine signatures were rejected, 26% of all the signatures (genuine and skilled forgeries) were classified as uncertain. In other words, by classifying only 26% of the signatures as uncertain we were able to obtain 0% false reject rate and 15% false accept rate. Slightly better results were obtained using the envelopes (upper and lower) alone. In this case, 10% of the genuine signatures were rejected, 13% of skilled forgeries were accepted, and only 16% of all signatures were classified as uncertain. Using either of the feature set combinations (envelopes and projections or just envelopes) was equally sufficient to successfully reject all of the random forgeries.

Data set	*FRR*	*FAR*	*Uncertain*
G1	0%	-	20%
F1	-	15%	28%
F2	-	0%	0%

Table 3.2: Performance results of the off-line signature verification system using both the envelopes and the projection profiles in the feature vector.

Data set	*FRR*	*FAR*	*Uncertain*
G1	10%	-	20%
F1	-	13%	13%
F2	-	0%	0%

Table 3.3: Performance results of the off-line signature verification system using only upper and lower envelopes in the feature vector.

Performance results of existing systems were reported with respect to false accept and false reject error rates. To be able to compare the results of our system with those of the existing systems, our system was retrained to classy signatures into 2 classes (genuine and forgery). Using the envelopes alone, we obtained a 25% of error rate in rejecting skilled forgeries, and a 20% of error rate in accepting genuine signatures.

As was mentioned in 3.1, state-of-the-art performance of the skilled forgery detection is around 20% equal error rate and that of random forgery detection is about 0.2% equal error rate. Although our system was evaluated on a small test data set, promising results are obtained.

3.9 Summary

Inspired by good results obtained from the on-line signature verification system, same methodology was adopted for the off-line system. However, due to the less discriminative information in the off-line signatures, lower performance results were obtained. Major challenges of the problem are the variation within genuine signa-

tures and hardly perceptible differences between a forgery and a genuine signatures of a particular writer.

Image of a signatures, signed with a dark ink on a relatively white paper, is the input to the system. Gaussian filtering followed by morphological closing and opening operators are applied to remove possible noise from the signature image. Then, the image size is normalized to get rid of the scaling problem. Four different feature vectors are extracted from preprocessed image: upper and lower envelopes, horizontal and vertical projections profiles of the signature. Feature vectors are then merged to a single vector which is used for the signature dissimilarity calculation. As the feature vector is extracted, the off-line signature verification system follows the same methodology as on-line system do.

The dissimilarity between two signatures is established using dynamic programming algorithm. To verify a test signature, first it is aligned with the reference set signature of the claimed user, resulting the three distances. Then each of the distances is normalized by the corresponding reference set average. Three dimensional feature vector, with the normalized distances in it, is first projected on to the principal component, and then classified using the linear classifier. Since the main task of the system is to ease the work of human signature experts, signatures are classified into three classes: genuine, forgery, and an uncertain. This classification scheme reduces work of experts, since only a small portion of presented signatures is classified into the third group, while the rest of the signatures are reliable classified into the corresponding classes. This is necessary as it is impossible to find features that can reliable separate the two classes (genuine or forgery).

The system is trained using the validation data set. 20 people contributed to the test data set with a total of 100 signatures (genuine and skilled forgery). 10% of the genuine signatures were rejected by the system and only 13% of forgeries accepted, while 16% of the total amount classified as uncertain signatures. Obtained results are comparable to those of the state-of-the-art.

Chapter 4

Conclusions

In this thesis we have addressed the problem of handwritten signature verification. We have formulated signature verification as a two-class problem and approached it using standard pattern recognition techniques, improving the decision step of the previous approaches which are commonly based on heuristic methods. Two complete systems were presented: one for on-line and the other for off-line signature verification. Both systems had similar verification methodology and differed only in data acquisition and feature extraction modules.

Another important design decision was about not doing any preprocessing for the on-line signature verification system. We have decided that the advantage of not resampling, namely keeping all the dynamic information, significantly outweighs the advantages of resampling.

In comparing two signatures, it is essential to use robust features that would result in low dissimilarity score for genuine signatures and high one for forgeries. After studying and experimenting with a number of features, for the on-line signature verification we have decided to use x and y coordinate differences between two consecutive points, as they are more robust to local variations of the signatures.

Another improvement was in the design of the dynamic programming algorithm. Various parameters were optimized. Parameters were chosen such that small variations within genuine signatures were ignored, while forgeries were still detected. In all steps where parameters were set, we used the validation data to keep testing completely unbiased.

In verifying a test signature, the signature is aligned with all reference set signatures belonging to the claimed user, resulting in a number of dissimilarity scores:

distances to nearest, farthest and template reference signatures. In previous systems, only one of these distances, typically the distance to the nearest reference signature or the distance to a template signature, was chosen, in an ad-hoc manner, to classify the signature as genuine or forgery. Here we proposed a method to utilize all of these distances, treating them as features in a two-class classification problem, using standard pattern classification techniques. The distances are first normalized, resulting in a three dimensional feature space where genuine and forgery signature distributions are well separated. We experimented with the Bayes classifier, Support Vector Machines, and Linear classifier to classify a given signature into one of the two classes (forgery or genuine).

Test data sets of 620 on-line and 100 off-line signatures were constructed to evaluate performances of the two systems. Off-line signature data set is smaller because it was time consuming to collect and scan the data. Since it is very difficult to obtain real forgeries, we obtained skilled forgeries which were supplied by forgers who had access to the signature data to practice before forging. Here we proposed a new method to collect skilled forgeries: signatures to be forged are animated according to the time stamps of their sampling points, hence a forger was able to see both the shape and the dynamics of the signature. The online system had a 1.4% error in rejecting forgeries, while rejecting only 1.3% of genuine signatures.

As the reliability of the off-line signature verification is lower, even for forensic experts, we decided to make a third category of signatures. Genuine and forgery signatures, those which can be detected with high confidence, are separated to their corresponding classes, while only a small portion of signatures are being identified as uncertain and left for human inspection. An offline signature is easier to forge, hence the offline system's performance was lower: 10% of the genuine signatures were rejected by the system, 13% of forgeries were accepted, while 16% of total signature amount were delivered to the uncertain signatures class. The results for the online system show significant improvement over the state-of-the-art results, and the results for the offline system are comparable with the performance of experienced human examiners.

Appendix A

Additional Feature Distribution Graphs

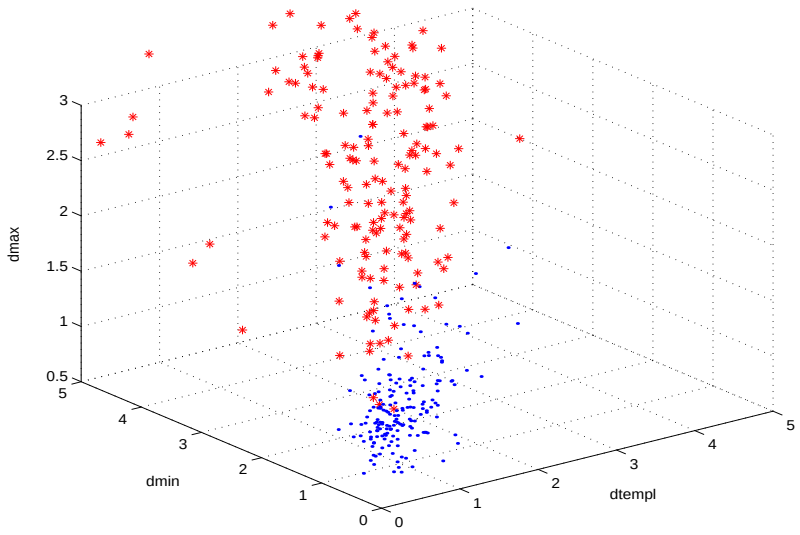

Figure A.1: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional distance vector, where calculation of distances is based on the x and y coordinates relative to the first point of a signature trajectory. dmax, dmin, and dtempl represent dimensions spanned by the corresponding normalized distances.

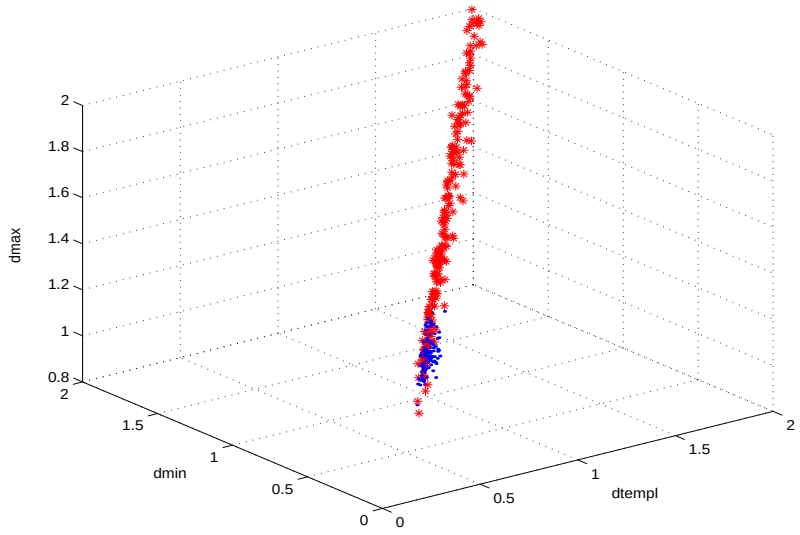

Figure A.2: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional distance vector, where calculation of distances is based on the curvature differences between two consecutive points of a signature trajectory. dmax, dmin, and dtempl represent dimensions spanned by the corresponding normalized distances.

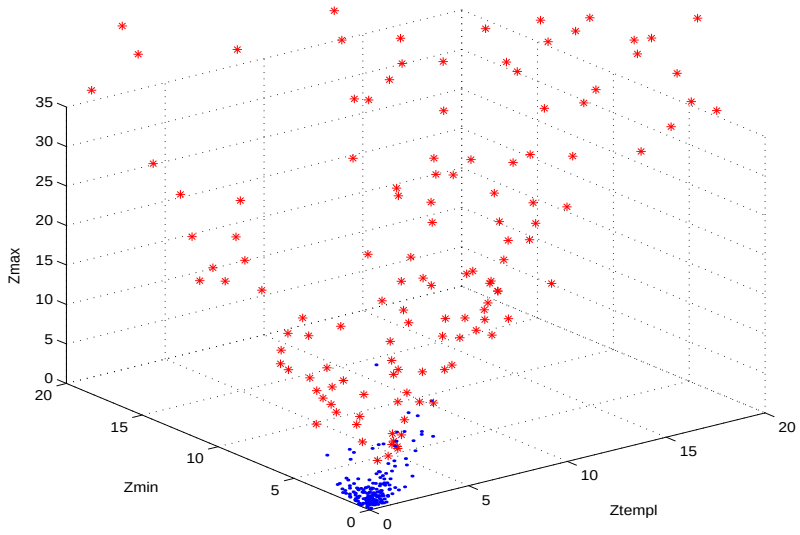

Figure A.3: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional z-score vector, where calculation of z-scores is based on the x and y coordinate differences between two consecutive points of a signature trajectory. zmax, zmin, and ztempl represent dimensions spanned by the corresponding z-scores.

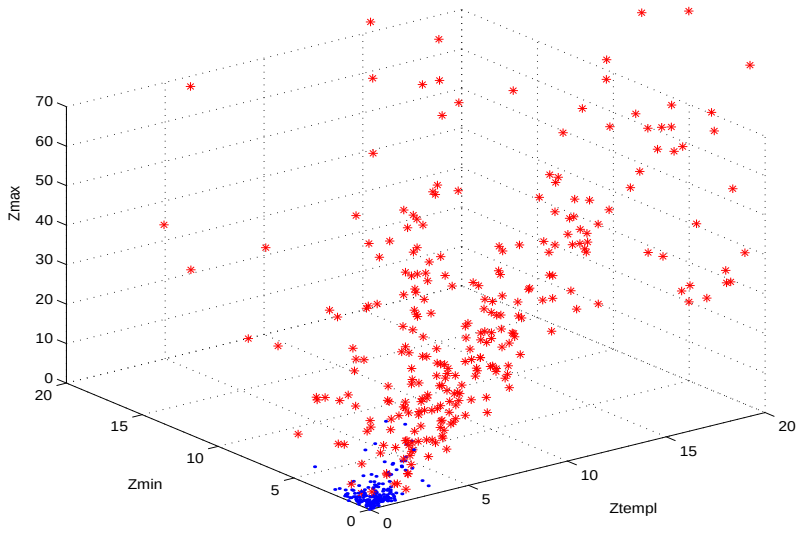

Figure A.4: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional z-score vector, where calculation of z-scores is based on the x and y coordinates relative to the first point of a signature trajectory. zmax, zmin, and ztempl represent dimensions spanned by the corresponding z-scores.

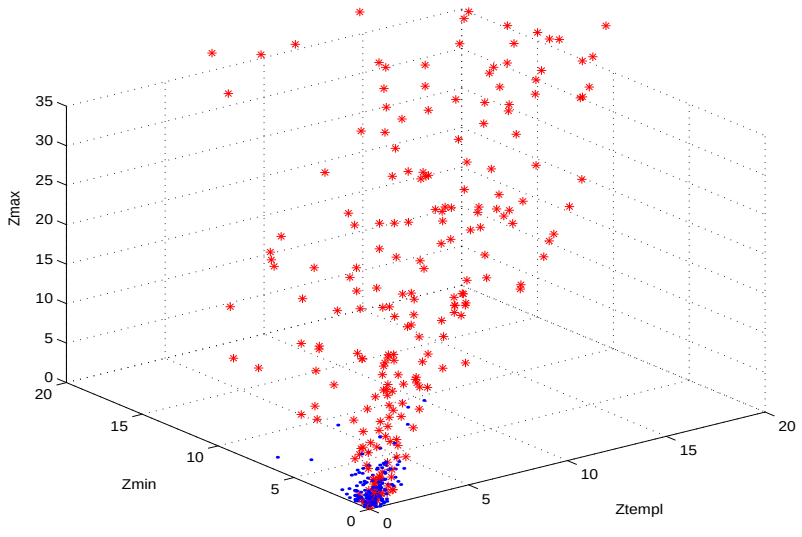

Figure A.5: Plot of genuine (blue dots) and forgery signatures (red stars) with respect to the 3-dimensional z-score vector, where calculation of z-scores is based on the curvature differences between two consecutive points of a signature trajectory. zmax, zmin, and ztempl represent dimensions spanned by the corresponding z-scores.

Appendix B

Additional System Performance Evaluation Results

Data Set	*Signature #*	*Type*
G1	182	Genuine
F1	313	Skilled forgeries
G2	124	Genuine

Table B.1: Data sets used to evaluate on-line system performance.

Classifier	*G1: FRR*	*F1: FAR*	*G2: FRR*	*Overall Error Rate*
Linear	1.65%	8.74%	4.84%	5.98%
Bayes	1.09%	17.9%	4.03%	10.19%
SVM	0.55%	13.15%	3.33%	7.45%
PCA on Z-Scores	2.74%	6.73%	13.71%	6.96%
Bayes on Z-Scores	1.09%	16.66%	7.25%	10.19%

Table B.2: System performance results using the classifiers mentioned in section 2.8 and x and y coordinates relative to the first point of a signature trajectory in feature vectors.

Classifier	*G1: FRR*	*F1: FAR*	*G2: FRR*	*Overall Error Rate*
Linear	2.74%	13.46%	8.06%	9.22%
Bayes	6.04%	14.10%	8.06%	10.51%
SVM	0%	15.70%	3.33%	8.58%
PCA on Z-Scores	2.74%	14.74%	6.45%	9.54%
Bayes on Z-Scores	6.04%	13.14%	8.06%	10.03%

Table B.3: System performance results using the classifiers mentioned in section 2.8 and curvature differences between two consecutive points in feature vectors.

Bibliography

[1] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol. 2, pp. 121-167, 1998.

[2] D. J. Burr, "Experiments With a Connectionist Text Reader", IEEE International Conference on Neural Networks, San Diego, CA, pp. 717-724, 1987.

[3] P. C. Chuang, "Machine Verification of Handwritten Signature Image", In Proceedings of International Conference on Crime Countermeasure, pp. 105-109, 1977.

[4] N. Cristianini, and B. Scholkopf, "Support Vector Machines and Kernel Methods, The New Generation of Learning Machines", AI Magazine, Vol. 23, No. 3, pp. 31-41, 2002.

[5] H. Dolfing, E. Aarts, and van J.J. Oosterhout "On-Line Signature Verification with Hidden Markov Models", ICPR, 1998.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification", Second Edition, John Wiley & Sons, Inc., 2001.

[7] Evett and R. N. Totty, "Study Of The Variation In The Dimensions Of Genuine Signatures", Journal of the Forensic Science Society, vol. 25, pp. 207-215, 1985.

[8] B. Fang, C. H. Leung, Y. Y. Tang, K. W. Tse, P. C. K. Kwok, Y. K. Wong, "Off-Line Signature Verification by The Tracking of Feature and Stroke Positions", Pattern Recognition, Vol. 36, pp. 91-101, 2003

[9] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Addison Wesley, 1993.

[10] J. K. Guo, D. Doermann, and Azriel Rosenfeld, "Forgery Detection by Local Correspondence", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, pp. 579-641, 2001.

[11] B. Herbst, D. Richards, "On an Automated Signature Verification System", In Proceedings of IEEE International Symposium of Industrial Electronics, pp. 600-604, 1998.

[12] B. M. Herbst and H. Coetzer, "On an Off-line Signature Verification System", Proceedings of the 9th Annual South African Workshop on Pattern Recognition, pp. 39-43, 1998.

[13] A.K. Jain, F.D. Griess, and S.D. Connell, "On-line Signature Verification", Pattern Recognition, Vol. 35, pp. 2963-2972, Dec. 2002.

[14] I. T. Jollife, "Principal Component Analysis", NY: Springer Verlag, 1986.

[15] R. Martens, and L. Claesen, "Dynamic Programming Optimisation for On-Line Signature Verification", ICDAR97, 1997.

[16] R. Martens, and L. Claesen, "On-Line Signature Verification by Dynamic Time-Warping", Proceedings of the 13'th International Conference on Pattern Recognition, pp. 38-42, 1996.

[17] T. Matsuura, and H. Sakai, "On Stochastic Representation of Handwriting Process and Its Application to Signature Verification", Proceedings of ICSP'96, 1996.

[18] Y. Mizukami, H. Miike, M. Yoshimura, and I. Yoshimura, " An Off-Line Signature Verification System Using an Extracted Displacement Function", In Proceedings of ICDAR, pp. 757-760, 1999.

[19] M.E. Munich, and P. Perona, "Visual signature verification using affine arc-length", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.

[20] N. A. Murshed, F. Bortolozzi and R. Sabourin, "Off-Line Signature Verification, Without a Priori Knowledge of Class ω_2. A New Approach", ICDAR, Vol. 1, 1995.

[21] V. S. Nalwa, "Automatic On-Line Signature Verification", Proceedings of IEEE, vol. 85, pp. 215-239, 1997.

[22] W. F. Nemcek and W. C. Lin, "Experimental Investigation of Automatic Signature Verification" IEEE Transactions on Systems, Man and Cybernetics, Vol. 4, pp. 121-126, 1974.

[23] T. Ohishi, Y. Komiya, and T. Matsumoto, "On-line Signature Verification Using Pen-position, Pen-pressure and Pen-inclination Trajectories", ICPR, Vol. IV, pp. 547-550, 2000.

[24] R. Plamondon, G. Lorette, "Automatic Signature Verification and Writer Identification-The State of the Art", Pattern Recognition vol. 22, pp. 107-131, 1989.

[25] M. J. Revillet, "Signature Verification on Postal Cheques", in Proceedings of ICDAR, pp. 767-773, 1991

[26] G. Rigoll, A. Kosmala, "A Systematic Comparison Between On-Line and Off-Line Methods for Signature Verification with Hidden Markov Models", In Proceedings of International Conference on Pattern Recognition, vol. 2, pp. 1755-1757, 1998.

[27] R. Sabourin and G. Genest, "An Extended-Shadow-Code Based Approach for Off-Line Signature Verification" ICDAR, 1993.

[28] R. Sabourin, G. Genest, and F. Preteux, "Pattern Spectrum as a Local Shape Factor for Off-Line Signature Verification", In Proceedings of ICPR, Vol. 3 pp. 43-48, 1996.

[29] R. Sabourin, G. Genest, and F. Preteux, "Off-Line Signature Verification by Local Granulometric Size Distributions", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 9, 1997.

[30] R. Sabourin, J. P. Drouhard, and E. S. Wah "Shape Matrices as a Mixed Shape Factor for Off-line Signature Verification", In Proceedings of ICDAR, pp. 661-664, 1997.

[31] B. Scholkopf, C. J. C. Burges, and A. J. Smola, "Advances in Kernel Methods: Support Vector Learning", MA: MIT Press, Cambridge, 1999.

[32] Smart Computing, "Digital Design Past, Present & Future Of Digital Tablets", Oct., 2002, Vol. 6, Issue 8, pp. 36-39.

[33] T. S. Tolba, "GloveSignature: A Virtual-Reality-Based System for Dynamic Signature Verification", Digital Signal Processing Vol. 9, pp. 241-266, 1999. (article available online at http://www.idealibrary.com)

[34] V. Vapnik, "Statistical Learning Theory", NY: Wiley, New York, 1998.

[35] C. Vielhauer, R. Steinmetz, and A. Mayerhofer, "Biometric Hash Based on Statistical Features of On-line Signatures", 16'th International Conference on Pattern Recognition, 2002.

[36] X. Yang, T. Furuhashi, K. Obata, and Y. Uchikawa, "Constructing a High Performance Signature Verification System Using a GA Method", 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems (ANNES '95), 1995.

Bibliography

[1] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol. 2, pp. 121-167, 1998.

[2] D. J. Burr, "Experiments With a Connectionist Text Reader", IEEE International Conference on Neural Networks, San Diego, CA, pp. 717-724, 1987.

[3] P. C. Chuang, "Machine Verification of Handwritten Signature Image", In Proceedings of International Conference on Crime Countermeasure, pp. 105-109, 1977.

[4] N. Cristianini, and B. Scholkopf, "Support Vector Machines and Kernel Methods, The New Generation of Learning Machines", AI Magazine, Vol. 23, No. 3, pp. 31-41, 2002.

[5] H. Dolfing, E. Aarts, and van J.J. Oosterhout "On-Line Signature Verification with Hidden Markov Models", ICPR, 1998.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification", Second Edition, John Wiley & Sons, Inc., 2001.

[7] Evett and R. N. Totty, "Study Of The Variation In The Dimensions Of Genuine Signatures", Journal of the Forensic Science Society, vol. 25, pp. 207-215, 1985.

[8] B. Fang, C. H. Leung, Y. Y. Tang, K. W. Tse, P. C. K. Kwok, Y. K. Wong, "Off-Line Signature Verification by The Tracking of Feature and Stroke Positions", Pattern Recognition, Vol. 36, pp. 91-101, 2003

[9] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Addison Wesley, 1993.

[10] J. K. Guo, D. Doermann, and Azriel Rosenfeld, "Forgery Detection by Local Correspondence", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, pp. 579-641, 2001.

[11] B. Herbst, D. Richards, "On an Automated Signature Verification System", In Proceedings of IEEE International Symposium of Industrial Electronics, pp. 600-604, 1998.

[12] B. M. Herbst and H. Coetzer, "On an Off-line Signature Verification System", Proceedings of the 9th Annual South African Workshop on Pattern Recognition, pp. 39-43, 1998.

[13] A.K. Jain, F.D. Griess, and S.D. Connell, "On-line Signature Verification", Pattern Recognition, Vol. 35, pp. 2963-2972, Dec. 2002.

[14] I. T. Jollife, "Principal Component Analysis", NY: Springer Verlag, 1986.

[15] R. Martens, and L. Claesen, "Dynamic Programming Optimisation for On-Line Signature Verification", ICDAR97, 1997.

[16] R. Martens, and L. Claesen, "On-Line Signature Verification by Dynamic Time-Warping", Proceedings of the 13'th International Conference on Pattern Recognition, pp. 38-42, 1996.

[17] T. Matsuura, and H. Sakai, "On Stochastic Representation of Handwriting Process and Its Application to Signature Verification", Proceedings of ICSP'96, 1996.

[18] Y. Mizukami, H. Miike, M. Yoshimura, and I. Yoshimura, " An Off-Line Signature Verification System Using an Extracted Displacement Function", In Proceedings of ICDAR, pp. 757-760, 1999.

[19] M.E. Munich, and P. Perona, "Visual signature verification using affine arc-length", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.

[20] N. A. Murshed, F. Bortolozzi and R. Sabourin, "Off-Line Signature Verification, Without a Priori Knowledge of Class ω_2. A New Approach", ICDAR, Vol. 1, 1995.

[21] V. S. Nalwa, "Automatic On-Line Signature Verification", Proceedings of IEEE, vol. 85, pp. 215-239, 1997.

[22] W. F. Nemcek and W. C. Lin, "Experimental Investigation of Automatic Signature Verification" IEEE Transactions on Systems, Man and Cybernetics, Vol. 4, pp. 121-126, 1974.

[23] T. Ohishi, Y. Komiya, and T. Matsumoto, "On-line Signature Verification Using Pen-position, Pen-pressure and Pen-inclination Trajectories", ICPR, Vol. IV, pp. 547-550, 2000.

[24] R. Plamondon, G. Lorette, "Automatic Signature Verification and Writer Identification-The State of the Art", Pattern Recognition vol. 22, pp. 107-131, 1989.

[25] M. J. Revillet, "Signature Verification on Postal Cheques", in Proceedings of ICDAR, pp. 767-773, 1991

[26] G. Rigoll, A. Kosmala, "A Systematic Comparison Between On-Line and Off-Line Methods for Signature Verification with Hidden Markov Models", In Proceedings of International Conference on Pattern Recognition, vol. 2, pp. 1755-1757, 1998.

[27] R. Sabourin and G. Genest, "An Extended-Shadow-Code Based Approach for Off-Line Signature Verification" ICDAR, 1993.

[28] R. Sabourin, G. Genest, and F. Preteux, "Pattern Spectrum as a Local Shape Factor for Off-Line Signature Verification", In Proceedings of ICPR, Vol. 3 pp. 43-48, 1996.

[29] R. Sabourin, G. Genest, and F. Preteux, "Off-Line Signature Verification by Local Granulometric Size Distributions", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 9, 1997.

[30] R. Sabourin, J. P. Drouhard, and E. S. Wah "Shape Matrices as a Mixed Shape Factor for Off-line Signature Verification", In Proceedings of ICDAR, pp. 661-664, 1997.

[31] B. Scholkopf, C. J. C. Burges, and A. J. Smola, "Advances in Kernel Methods: Support Vector Learning", MA: MIT Press, Cambridge, 1999.

[32] Smart Computing, "Digital Design Past, Present & Future Of Digital Tablets", Oct., 2002, Vol. 6, Issue 8, pp. 36-39.

[33] T. S. Tolba, "GloveSignature: A Virtual-Reality-Based System for Dynamic Signature Verification", Digital Signal Processing Vol. 9, pp. 241-266, 1999. (article available online at http://www.idealibrary.com)

[34] V. Vapnik, "Statistical Learning Theory", NY: Wiley, New York, 1998.

[35] C. Vielhauer, R. Steinmetz, and A. Mayerhofer, "Biometric Hash Based on Statistical Features of On-line Signatures", 16'th International Conference on Pattern Recognition, 2002.

[36] X. Yang, T. Furuhashi, K. Obata, and Y. Uchikawa, "Constructing a High Performance Signature Verification System Using a GA Method", 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems (ANNES '95), 1995.