UNIVERSITY OF WEST BOHEMIA IN PILSEN
—
Faculty of Applied Science
Department of Computer Science and Engineering
Laboratory of Intelligent Communication Systems

Ondřej Rohlík

Handwritten Text Analysis

Thesis

(Computer Science)

Supervised by Václav Matoušek

Pilsen March 2003

Abstract

Automatic handwritten text recognition is an extremely active research area. Although systems exist on the market, there are few that can promise sufficiently high accuracy rates at a reasonable level of efficiency. This thesis focuses on automatic handwritten signature verification—widely accepted non-invasive biometrics with considerable legal recognition and wide current usage in document authentication and transaction authorization.

An overview of automated biometric systems is provided. An introduction to the methods of biometric system evaluation is presented and is followed by a comparison of the most frequently used biometrics.

The state-of-the-art automatic handwriting analysis systems—both the handwritten text recognition systems and the handwritten signature verification systems—are briefly described and new trends in off-line and on-line handwriting processing are discussed.

The main part of the thesis deals with the unique biometrical pen designed for on-line data acquisition. The pen measures both the trajectory of the pen and the pressure of the pen nib on the paper. As the data acquired from the pen are entirely different from the data produced by other systems currently available, new handwritten analysis methods have to be developed. Some of the methods that have already been implemented are mentioned and their accuracy rates are discussed.

The aims of a future doctoral thesis are sketched at the end of this work. The first includes the design, implementation and evaluation of handwritten signature verification system based on the new biometric pen which utilizes both the best signature verification methods available. The second aim is to propose a highly sophisticated methodology of the verification process.

Contents

(1) Introduction

While information becomes a much more valuable commodity, the protection of that information becomes much more important. As a result, there is considerable interest in the development of an effective electronic user authentication technique which is not as intrusive and privacy invasive as identity verification based on fingerprints or DNA.

Handwritten signature verification appears to be a superior method because it is certainly more reliable for authentication purposes than, e.g., the use of a password or a personal identification number (PIN), since it is much more difficult to forge an individual's signature (both shape and dynamics) than it is to perhaps guess (or steal by some means of observation) a person's hand-typed password or PIN. It also complies with the ethical side of personal authentication as it is generally not considered an invasion of privacy or overly intrusive to store an individual's signature. Moreover, a handwritten signature is an identity verification method that has been used for centuries with considerable legal recognition and wide current usage in document and transaction authorization.

Signature verification is a domain that is highly specialized but at the same time very general. On one hand, the practical objective is clear and precise: to develop a system capable of verifying the identity of an individual based on an analysis of his or her signature through a process that distinguishes a genuine signature from a forgery. On the other hand, the achievement of this objective requires that the researcher takes an interest in many related domains: human-computer interaction, digital signal and image processing, system security, knowledge representation, psychophysics of perception and movement, forensic sciences, etc.

A signature verification system based on a unique-instrumented-pen is being developed by the Laboratory of Intelligent Communication Systems at the University of West Bohemia in close cooperation with the University of Applied Sciences in Regensburg. The experience gathered from more than five years of active research is presented in this thesis.

The thesis is organized as follows. Chapter 2 discusses the advantages of automated biometric-based authentication and provides a survey of the six most popular biometrics. Chapter 3 gives an overview of handwriting, its nature, generation and perception. Furthermore, several notes on how human signature experts verify handwritten signatures are discussed. Chapter 4 describes state-of-the-art automatic handwriting analysis systems—both the handwritten text recognition systems and handwritten signature verification systems—and discusses new trends in off-line and on-line handwriting processing. Chapter 5 gives a condensed description of the innovative devices and methods already proposed and implemented together with a short evaluation of their performance and suitability for further application in the automatic analysis of handwriting. Finally the author's doctoral thesis objectives are sketched.

(2) Biometrics

2.1 Biometrics and Its Applications in Person Identification

In the modern networked society, there is an ever-growing need to determine or verify the identity of a person. Where authorization is necessary for any action, be it picking up a child from daycare or boarding an aircraft, authorization is almost always vested in a single individual or a class of individuals.

There is a number of methods to verify identity adopted by society or automated systems. These are summarized in Table 2.1. Traditional existing methods can be grouped into three classes [77]: (i) possessions; (ii) knowledge and (iii) biometrics. *Biometrics* is the science of identifying or verifying the identity of a person based on physiological or behavioral characteristics. Physiological characteristics include fingerprints and facial image. The behavioral characteristics are actions carried out by a person in a characteristic way and include signature and voice, though these are naturally dependent on physical characteristics. The three identification methods are often used in combination, e.g.: the possession of a key is a physical conveyor of authorization; a password plus a user ID is a purely knowledge-based method of identification; an ATM card is a possession that requires knowledge to carry out a transaction; a passport is a possession that requires biometric verification (facial image and signature).

Method	Examples	Comments
What you have	Cards, badges, keys	Can be lost or stolen
What you know	User ID, password, PIN	Can be shared Can be duplicated Can be forgotten
What you are	Fingerprint, face	Can be shared Can be guessed Non-repudiable authentication

Tab. 1: Identification technologies

Early automated authorization and authentication methods relied on possessions and knowledge, however, there are several well-known problems associated with these methods that restrict their use and the extent to which they can be trusted. These methods verify attributes which are usually assumed to imply the presence of a given person. The most important drawbacks of these methods are that (i) possessions can be lost, forged or easily duplicated; (ii) knowledge can be forgotten; (iii) both knowledge and possessions can be shared or stolen.

Clearly, this cannot be tolerated in applications such as high security physical access control, bank account access and credit card authentication. The science of biometrics provides an elegant solution to these problems by truly verifying the identity of the individual.

For contemporary applications, biometric authentication is automated to eliminate the need of human verification, and a number of new biometrics have been developed, taking advantage of improved understanding of the human body and advanced sensing techniques [51]. New physiological biometric authentication technologies that have been developed include iris patterns, retinal images and hand geometry. New behavioral biometrics technologies, still very much in the research stage, are gait and key stroke patterns.

A biometrics system works with an enrolled biometric (identity) which is the first step. After enrolling, the user can be verified many times.

The behavioral characteristics must be insensitive to variations due to the state of health, mood of the user or passage of time. The physiological characteristics remain fairly constant over time.

Basically, there are two types of application scenarios: *identification* and *authentication*. For identification, also known as 1:N matching, the system uses the biometric to determine the corresponding person from a database containing many identities, or decides that a particular subject is not enrolled in the database. For authentication, also known as 1:1 matching or identity *verification*, the system matches the input biometric against a single biometric record that can be stored on an identification card presented at the transaction time or retrieved from a database with the help of a key such as an account number. The output is either "Yes" if the two biometrics match or "No" otherwise. During the enrollment process there is often employed an identification system to ensure that the subject is not already enrolled.

2.2 Pattern Recognition-Based Biometric Systems

Biometric systems can be considered as a generic pattern recognition system as shown in Fig. 2.1. The input subsystem consists of a special sensor needed to acquire the biometric signal. Reliable acquisition of the input signal is a challenge for sensor designers, especially in light of interpersonal and intrapersonal variations and varying environmental situations. The signal contains (in its raw form) the required identifying information hidden among irrelevant information. Invariant features are extracted from the signal for representation purposes by the feature extraction subsystem. During the enrollment process, a representation (called template) of the biometrics in terms of these features is stored in the system. The matching subsystem accepts query and reference templates and returns the degree of match or mismatch as a score, i.e., a similarity measure. A final decision step compares the score to a decision threshold to deem the comparison a match or non-match. The overall performance of the system depends on the performance of all the subsystems. In addition, the system designer has to focus on efficient storage and retrieval, error free transmission and possible encryption and decryption of the result as well as intermediate signals.

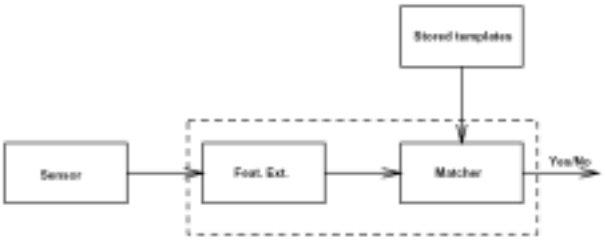

Figure 2.1: A generic biometric system

2.2.1 Classification Errors and Performance Evaluation

To assess the performance of a biometric system, it is possible to analyze it in a hypothesis testing framework. Let B' and B denote biometrics, e.g., two fingers. Further, let the stored biometric sample or template be pattern $P' = S(B')$ and the acquired one be pattern $P = S(B)$. Then, in terms of hypothesis testing, we have the null and alternative hypotheses:

$$\begin{array}{lll} H_0: & B = B', & \textit{the claimed identity is correct} \\ H_1: & B \neq B', & \textit{the claimed identity is not correct.} \end{array} \tag{2.1}$$

Often some similarity measure $s = Sim(P, P')$ is defined and H_0 is decided if $s \geq T_d$ and H_1 is decided if $s < T_d$, with T_d a decision threshold. Some systems use a distance or dissimilarity measure. Anyhow we assume a similarity measure without loss of generality.

2.2.2 Measures of Performance

The measure s is also referred to as a score. When $P = P'$, s is referred to as a *match score* and B and B' are called a *matched pair*. When $P \neq P'$, s is refered to as *non-match score* and B and B' are called *non-matched pair*.

For expression 2.1, deciding H_0 when H_1 is true gives a false acceptance; deciding H_1 when H_0 is true results in a false rejection. The False Accept Rate (FAR) (proportion of non-mated pairs resulting in false acceptance) and False Reject Rate (FRR) (proportion of mated pairs resulting in false rejection) together characterize the accuracy of a recognition system for a given decision threshold. Varying the threshold T_d trades FAR off against FRR. In Figure 2.2, the FAR is the area under the H_1 density function to the right of the threshold and the FRR is the area under the H_0 density function to the left of the threshold. More specifically for biometric systems, we can express the two errors as False Match Rate (FMR) and False Non-Match Rate (FNMR) [124].

The Equal Error Rate (EER) is the point at some threshold (T_{ERR}) where $FRR = FAR$, i.e., where the areas marked under the two curves in Fig. 2.2 are equal.

Rather than showing the error rates in terms of probability densities as in Figure 2.2, it is desirable to report system accuracy using a *Receiver Operating Curve* (ROC) [32], [87]. A ROC is a mapping $T_d \rightarrow (FAR, FRR)$,

$$ROC(T_d) = (FAR(T_d), FRR(T_d)),$$

as shown in Fig. 2.3.

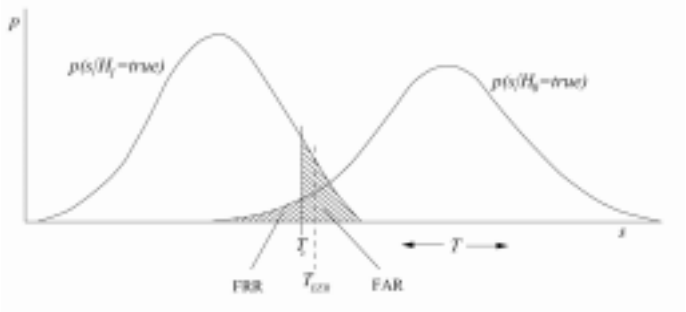

Figure 2.2: Impostor and genuine distributions with classification error definitions

Note that in a typical recognition system, all the information contained in the probability density functions (PDF) is also contained in the ROC. The ROC can be directly constructed from the PDFs as

$$FAR(T_d) = Prob(s \geq T_d \mid H_1 = true) = [\ 1 - \int_0^{T_d} p(s \mid H_1 = true)\, \mathrm{d}s\ ]$$

$$FRR(T_d) = Prob(s < T_d \mid H_0 = true) = \int_0^{T_d} p(s \mid H_0 = true)\, \mathrm{d}s\ .$$

If T_d goes to zero, the FAR goes to one and the FRR goes to zero; if T_d goes to T_{max}, the FAR goes to zero and the FRR goes to one.

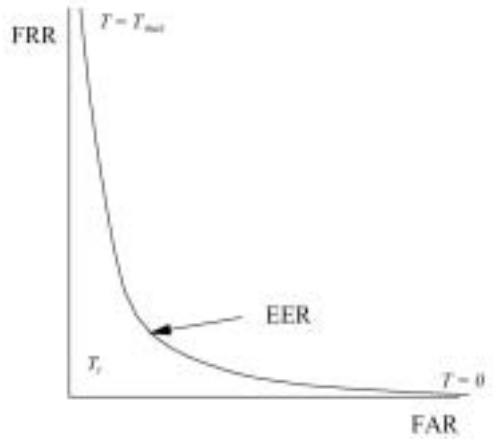

Figure 2.3: Receiver operating curve (ROC)

2.3 Survey of Commonly Used Biometrics

A brief description of the most widely used biometrics is provided in this section.

2.3.1 Fingerprint

Fingerprint is one of the most widely used biometrics. The advent of several inkless fingerprint scanning technologies coupled with the exponential increase in processor performance has taken fingerprint recognition beyond criminal identification applications to several civilian applications such as access control; time and attendance; and computer user login. Over the last decade, many novel techniques have been developed to acquire fingerprints without the use of ink. These scanners are known as *livescan* fingerprint scanners. The basic principle of these inkless methods is to sense the ridges on a finger, which are in contact with the surface of the scanner. The livescan image acquisition systems are based on four types of technology: Frustrated total internal reflection (FTIR) and other optical methods [36], CMOS capacitance [52], thermal [68] and ultrasound [5]. Recently, non-contact [134] fingerprint scanners have been announced that avoid problems related to touch-based sensing methods, including elastic distortion of the skin pattern.

The most commonly used fingerprint features are ridge bifurcations and ridge endings, collectively known as minutiae, which are extracted from the acquired image [74]. The feature extraction process starts by examining the quality of the input gray-level image. Virtually every published method of feature extraction [69], [102] computes the orientation field of the fingerprint image which reflects the local ridge direction at every pixel. The local ridge orientation has been used to tune filter parameters for enhancement and ridge segmentation. From the segmented ridges, a thinned image is computed to locate the minutiae features. Usually, one has to go through a minutia post-processing stage to clean up several spurious minutiae resulting from either enhancement, ridge segmentation or thinning artifacts. The main goal of the fingerprint authentication module is to report some sort of distance between two fingerprint feature sets accurately and reliably. The authentication function has to compensate for (i) translation, (ii) rotation, (iii) missing features, (iv) additional features, (v) spurious features and, more importantly, (vi) elastic distortion between a pair of feature sets.

Often storage and transmission of fingerprint images involves compression and decompression of the image. Standard compression techniques often remove the high frequency areas around the minutia features. Therefore a novel fingerprint compression scheme called *wavelet scalar quantization* (WSQ) is recommended by the FBI.

The main advantages of fingerprint as a biometric is the high accuracy and low cost of the system.

2.3.2 Iris

Although iris [126] is a relatively new biometric, it has been shown to be very accurate and stable. The colored part of the eye bounded by the pupil and sclera is the iris and is extremely rich in texture. Like fingerprints, this biometric results from the developmental process and is not dictated by genetics. So far, there has been only a couple off iris recognition systems described in the literature. The primary reason is the difficulty of designing a reliable image acquisition device. Often iris recognition is confused with the *retinal recognition* system which has a much harder-to-use input acquisition subsystem. In [22] the texture of the iris is represented using Gabor wavelet responses and the matcher

is an extremely simple—fast Hamming distance measure.

2.3.3 Hand Geometry

Hand geometry based authentication is a limited *scalable*[1] but extremely user-friendly biometric. The lengths of the fingers and other hand shape attributes are extracted from images of a hand and used in the representation. To derive such gross characteristics, a relatively inexpensive camera can be employed resulting in an overall low cost system. As the computation is fairly light weight, a stand-alone system is easy to build. Moreover, this biometrics is not seen to compromise user privacy, it is quite widely accepted. However, hand geometry based authentication systems have relatively high FAR and FRR.

2.3.4 Face Recognition

Face recognition [15], [110] is a particularly compelling biometric because it is one used every day by nearly everyone on earth. Since the advent of photography it has been institutionalized as a guarantor of identity in passports and identity cards. Because faces are easily captured by conventional optical imaging devices, there are large legacy databases (police mug-shots and television footage, for instance) that can be automatically searched. Because of its naturalness, face recognition is more acceptable than most biometrics, and the fact that cameras can acquire the biometric passively means that it can be very easy to use. Indeed, surveillance systems rely on capturing the face image without the cooperation of the person being imaged.

Despite these attractions, face recognition is not sufficiently accurate to accomplish the large-population identification tasks (if compared with fingerprint or iris). One clear limit is the similarity of appearance of identical twins, but determining the identity of two photographs of the same person is hindered by all of the following problems, which can be divided into three classes:

- Physical changes: expression change; aging; personal appearance (make-up, glasses, facial hair, hairstyle, disguise).
- Acquisition geometry changes: change in scale, location and in-plane rotation of the face (facing the camera) as well as rotation in depth (facing the camera obliquely).
- Imaging changes: lighting variation; camera variations; channel characteristics (especially in broadcast, or compressed images).

No current system can claim to handle all of these problems well. Indeed there has been little research on making face recognition robust to aging. In general, constraints on the problem definition and capture situation are used to limit the amount of invariance that needs to be afforded algorithmically.

The main challenges of face recognition today are handling rotation in depth and broad lighting changes, together with personal appearance changes. There is interest in other acquisition modalities such as 3D shape through stereo or range-finders; near infrared or facial thermograms, all of which have attractions, but lack the compelling reasons for visible-light face recognition outlined above.

[1] Scalable means that the performance degrades slowly as the database size increases.

In general, face recognition systems proceed by detecting the face in the scene, thus estimating and normalizing for translation, scale and in-plane rotation. The two approaches to face recognition are [10] appearance-based and geometric approach, analyzing the appearance of the face and the distances between features respectively. In many systems these are combined, and indeed to apply appearance-based methods in the presence of facial expression changes requires generating an expressionless "shape-free" face by image warping. Appearance based methods can be global, where the whole face is considered as a single entity, or local, where many representations of separate areas of the face are created.

Considerable progress has been made in recent years, with much commercialization of face recognition, but a lot remains to be done towards the "general" face recognition problem.

2.3.5 Speaker Identification

Like face recognition, speaker identification [37] has attractions because of its prevalence in human communication. We expect to pick up the phone and be able to recognize someone by their voice after only a few words, although clearly the human brain is very good at exploiting context to narrow down the possibilities. Telephony is the main target of speaker identification, since it is a domain with ubiquitous existing hardware where no other biometric can be used. Increased security for applications such as telephone banking means that the potential for deployment is very large. Speaking solely in order to be identified can be somewhat unnatural, but in situations where the user is speaking anyway (e.g., a voice-controlled computer system, or when ordering something by phone) the biometric authentication becomes "passive". If a video signal is available lip-motion identification can also be used [28], [55], [66].

Speaker identification suffers considerably from any variations in the microphone [45], [104] and transmission channel, and performance drop badly when use conditions are mismatched. Background noise can also be a considerable problem in some circumstances, and variations in voice due to illness, emotion or aging are further problems that have received little study.

Speaker verification is particularly vulnerable to *replay attacks* because of the ubiquity of sound recording and play-back devices. Consequently more thoughts has been given in this domain to avoiding such attacks. We can categorize speaker identification systems depending on the freedom in what is spoken:

Fixed text: The speaker says a predetermined word or phrase, which was recorded at enrollment. The word may be secret, so acts as a password, but once recorded a replay attack is easy, and re-enrollment is necessary to change the password.

Text dependent: The speaker is prompted by the system to say a specific thing. The machine aligns the utterance with the known text to determine the user. For this, enrollment is usually longer, but the prompted text can be changed at will. Limited systems (e.g., just using digit strings) are vulnerable to splicing-based replay attacks.

Text independent: The speaker ID system processes any utterance of the speaker. Monitoring can be continuous—the more is said the greater the system's confidence in the identity of the user. The advent of trainable speech synthesis might enable attacks on this approach.

	Fingerprint	Speech	Face	Iris	Hand	Signature
Maturity	very high	high	medium	high	medium	medium
Best FAR	10^{-8}	10^{-2}	10^{-2}	10^{-10}	10^{-4}	10^{-4}
Best FRR	10^{-3}	10^{-3}	10^{-2}	10^{-4}	10^{-4}	10^{-4}
Scalability	high	medium	medium	very high	low	medium
Sensor cost	< \$100	< \$5	< \$50	< \$3000	< \$500	< \$100
Sensor size	small	very small	small	medium	large	medium
Data size	< 200B	< 2KB	< 2KB	256B	< 10B	< 200B

Table 2.1: Comparison of six popular biometrics.

2.3.6 Signature Verification

Signature verification [82] is another biometric that has a long pedigree before the advent of computers, with considerable legal recognition and wide current usage in document authentication and transaction authorization in the form of checks and credit card receipts. Here the natural division is on-line vs. off-line, depending on the sensing modality. Off-line or "static" signatures are scanned from paper documents where they were written in the conventional way. On-line or "dynamic" signatures are written with an electronically instrumented device and the dynamic information (pen tip location through time) is usually available at high resolution, even when the pen is not in contact with the paper. Some on-line signature capture systems can also measure pen angle and contact pressure [27]. These systems provide a much richer signal than is available in the off-line case, and make the identification problem correspondingly easier. These additional data make on-line signatures very robust to forgery. While forgery is a very difficult subject to research thoroughly (Section 3.4), it is widely believed that most forgery is very simple and can be prevented using even relatively simple algorithms.

Because of the need of the special hardware for the more robust on-line recognition, it may seem unlikely that signature verification would spread beyond the domains where it is already used, but the volume of signature authorized transactions today is huge, making automation through signature verification very important (Section 4.4).

Table 2.1 compares the six biometrics [103]. The comparison is based on the following factors: (i) maturity; (ii) accuracy; (iii) scalability; (iv) cost; (v) obtrusiveness; (vi) sensor size; and, (vii) representation (template) size.

While technologies continue to advance and new biometrics are being pioneered, it seems clear that the complementary features of different biometrics will cause that each finds its own domains of applicability, with no single biometric dominating the field.

(3) Handwriting Analysis

Handwriting is a skill that is personal to individuals. Fundamental characteristics of handwriting are threefold. It consists of artificial graphical marks on a surface; its purpose is to communicate something; this purpose is achieved by virtue of the mark's conventional relation to language [19]. Each script consists of a set of icons, which are known as characters or letters, that have certain basic shapes. There are rules for combining letters to represent shapes of higher level linguistic units. For example, there are rules for combining the shapes of individual letters so as to form cursively written words in the Latin alphabet.

3.1 Survival of Handwriting

Copybooks and various writing methods, like the Palmer method, handwriting analysis, and autograph collecting, are words that conjure up a lost world in which people looked to handwriting as both a lesson in conformity and a talisman of the individual [121]. The reason that handwriting persists in the age of the digital computer is the convenience of paper and pen as compared to keyboards for numerous day-to-day situations. Handwriting was developed a long time ago as a means to expand human memory and to facilitate communication.

At the beginning of the new millennium, technology has once again brought handwriting to a crossroads. Nowadays, there are numerous ways to expand human memory as well as to facilitate communication and in this perspective, one might ask: Will handwriting be threatened with extinction, or will it enter a period of major growth?

Handwriting has changed tremendously over time and each technology-push has contributed to its expansion. The printing press and typewriter opened up the world to formatted documents, increasing the number of readers that, in turn, learned to write and to communicate. Computer and communication technologies such as word processors, fax machines, and e-mail are having an impact on literacy and handwriting. Newer technologies such as personal digital assistants (PDAs) and digital cellular phones will also have an impact [96].

All these inventions have led to the fine-tuning and reinterpreting of the role of handwriting and handwritten messages. As a general rule, it seems that as the length of handwritten messages decreases, the number of people using handwriting increases [91]. The signature can be considered the the limit.

3.2 Handwriting Generation and Perception

The study of handwriting covers a very broad field dealing with numerous aspects of this very complex task. It involves research concepts from several

disciplines: experimental psychology, neuroscience, physics, anthropology, education, forensic document examination, etc. [96].

From a generation point of view, handwriting involves several functions. Starting from a communication intention, a message is prepared at the semantic, syntactic, and lexical levels and converted somehow into a set of allographs (letter shape models) and graphs (specific instances) made up of strokes so as to generate a pen nib trajectory that can be recorded on-line with an instrumented pen or a digitizer. In many cases, the trajectory is just recorded on paper and the resulting document can be read later with an off-line system (e.g. scanner).

The understanding of handwriting generation is important in the development of both on-line and off-line recognition systems, particularly in accounting for the variability of handwriting. So far, numerous models have been proposed to study and analyze handwriting. These models are generally divided into two major classes: top-down and bottom-up models [95]. Top-down models refer to approaches that focus on high-level information processing, from semantics to basic motor control problems. Most of the top-down models have been developed for language processing purposes. They are not exclusively dedicated to handwriting and deal with the integration of lexical, syntactic, and semantic information to process a message. Bottom-up models are concerned with the analysis and synthesis of low-level neuromuscular processes involved in the production of a single stroke, going upward to the generation of graphs, allographs, words, etc. [96].

From an opposite point of view, the reading of a handwritten document relies on a basic knowledge about perception [111], [114]. Psychological experiments in human character recognition show two effects: (i) a character that either occurs frequently, or has a simple structure, is processed as a single unit without any decomposition of the character structure into simpler units and (ii) with infrequently occurring characters, and those with complex structure, the amount of time taken to recognize a character increases as its number of strokes increases. The former method of recognition is referred to as *holistic* and the latter as *analytic*.

The perceptual processes involved in reading have been discussed extensively in the cognitive psychology literature [4], [117], [119]. Such studies are pertinent in that they can form the basis for algorithms that emulate human performance in reading [6], [20] or try to do better [115]. Although much of this literature refers to the reading of machine-printed text, some conclusions are equally valid for handwritten text. For instance, the saccade (eye movement) fixate at discrete points on the text, and at each fixation the brain uses the visual peripheral field to infer the shape of the text. Algorithmically, this again leads to the holistic approach to recognition [96].

3.3 Handwriting Types

The two main types of handwriting can be distinguished—handprinted words (Fig. 3.1) and cursive written words (Fig. 3.2).

Handprinted words

Considering its dificulties, handprinted script poses the least problems for automatic handwriting processing. Handprinted characters can be divided into

Figure 3.1: Hand printed words (boxed discrete characters)

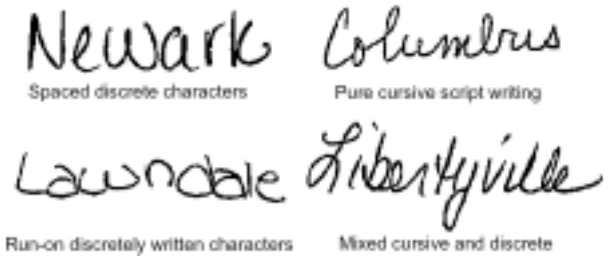

Figure 3.2: Types of cursive script

boxed discrete characters, spaced discrete characters, and run-on discrete characters. Figures 3.1 and 3.2 show examples for each type of writing. The example in Figure 3.1 shows an address that is written in boxed discrete characters, which are typical for checks and forms. The difficulty of automatic handwriting processing is strongly connected with the separation of letters. Separating the letters is called *character segmentation*.

Character segmentation simplifies the recognition process because the recognition of specific characters is less complex than the recognition of complete words. Discrete characters written in boxes require no character segmentation since the separation of characters has already been realized by the writer.

Spaced discrete characters and run-on discrete characters do require character segmentation. Spaced discrete characters are, however, easy to segment because every character corresponds to a connected component. Run-on discretely written characters consist of one or more strokes and segmentation can only occur after a stroke, i.e. after a pen lift. This type of handwriting is typically examined in on-line handwriting recognition.

Cursive written words

Cursive written words are harder to process than printed words because the separation of characters is more dificult. An example of a cursive written word is shown in Figure 3.2. Segmentation is more complicated for cursive writing since the characters of a word can overlap each other or can be written with one stroke, i.e. without lifting the pen.

Combinations of both types of handwriting are possible, i.e. some parts of a word can be printed and some can be cursively written.

3.4 Graphology and Handwriting Analysis

Graphology, the study of handwriting to determine one's personality traits, is not *handwriting analysis*. True handwriting analysis (as part of *document exam-*

ination) involves painstaking examination of the design, shape and structure of handwriting to determine authorship of a given handwriting sample. The basic principle underlying handwriting analysis is that no two people write the exact same thing the exact same way. Every person develops unique peculiarities and characteristics in their handwriting.

Handwriting analysis looks at letter formations, connecting strokes between the letters, upstrokes, retraces, down strokes, spacing, baseline, curves, size, distortions, hesitations and a number of other characteristics of handwriting. By examining these details and variations in a questioned sample and comparing them to a sample of known authorship, a determination can be made as the whether or not the authorship is genuine [131].

The exact definition of *forgery* is needed. Forgery is the false making or material altering of any writing with intent to defraud [131]. In document examination four basic types of forgery are distinguished: traced, simulation, freehand, and lifted (although different terminology is used in evaluation of automatic handwritten signature verification systems—Section 4.4.1). There are a few different ways to do *traced forgeries*: with overlays (as with tracing paper), transmitted light (as with a light board), tracing the indentations left in the page underneath the original writing, and tracing patterns of dots that outline the writing to be forged. *Simulation* involves the copying of writing from a genuine article; trying to imitate the handwriting of the original. *Freehand forgeries* are written with no knowledge of the appearance of the original. The final type is a lifted forgery, in which tape is used to lift off a signature, then place it on another document.

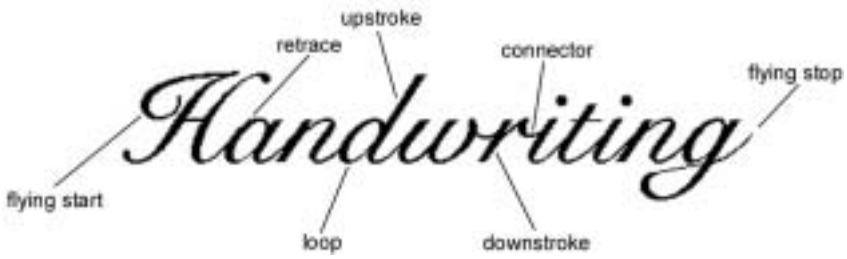

Figure 3.3: Examples of handwriting points of analysis

Freehand forgeries are the easiest to detect. Simulation forgeries are easy to detect for a number of reasons: it is very difficult to copy another's handwriting, the style is not fluent because the writing does not come naturally, and the forged writing will show tremors, hesitations, and other variations in letter quality that genuine handwriting would not have. Traced forgeries and lifts are easy enough to detect, but the identity of the forger cannot be determined.

3.5 Signature Analysis

Handwritten signatures come in many different forms and there is a great deal of variability even in signatures of people that use the same language. Some people simply write their name while others may have signatures that are only

vaguely related to their name and some signatures may be quite complex while others are simple and appear as if they may be forged easily [7]. It is also interesting that the signature style of individuals relates to the environment in which the individual developed their signature. For example, people in the United States tend to use their names as their signature whereas Europeans tend away from directly using their names. Systems which rely directly on the American style of signing may not perform as well when using signatures of Europeans, or signatures written in different languages [81].

It is well known that no two genuine signatures of a person are precisely the same and some signature experts note that if two signatures of the same person written on paper were identical they could be considered forgery by tracing. Successive signatures by the same person will differ, both globally and locally and may also differ in scale and orientation. In spite of these variations, it has been suggested that human experts are very good in identifying forgeries but perhaps not so good in verifying genuine signatures. For example, in [46] there are references cited indicating that as high as 25% genuine signatures were either rejected or classified as noopinion by trained document examiners while no forgeries were accepted (0% FAR and 25% FRR). Untrained personnel were found to accept up to 50% forgeries.

According to [86] handwriting shows great variation in speed and muscular dexterity. Forgeries vary in perfection all the way from the clumsy effort which anyone can see is spurious, up to the finished work of the adept which no one can detect. Experience shows that the work of the forger is not usually well done and in many cases is very clumsy indeed. The process of forging a signature or simulating another person's writing, if it is to be successful, involves a double process requiring the forger to not only copy the features of the writing imitated but must also hiding the writer's own personal writing characteristics. If the writing is free and rapid it will almost certainly show, when carefully analyzed, many of the characteristics of the natural writing of the writer no matter what disguise may have been employed.

Unusual conditions under which signatures are written may affect the signature. For example, hastily written, careless signatures, like those written in a delivery person's books, cannot always be used unless one has sample signatures that have been written under similar conditions. Furthermore, signatures written with a strange pen or in an unaccustomed place are likely to be different than the normal signatures of an individual[1]. When a signature is being written to be used for comparison this can also produce a selfconscious, unnatural signature. Variations in handwriting are themselves habitual and this is clearly shown in any collection of genuine signatures produced at different times and under a great variety of conditions, which when carefully examined show running through them a marked, unmistakable individuality even in the manner in which the signatures vary as compared with one another.

In [48] it is discussed what a signature is and how it is produced. Signature has at least three attributes: form, movement and variation, and since the signatures are produced by moving a pen on a paper, movement perhaps is the most important part of a signature. The movement is produced by muscles of the fingers, hand, wrist, and for some writers the arm, and these muscles are

[1]This has to be considered when designing specialized hardware devices (e.g. electronic pens) for signature capturing.

controlled by nerve impulses. Once a person is used to signing his or her signature, these nerve impulses are controlled by the brain without any particular attention to detail. Furthermore, a person's signature does evolve over time and with the vast majority of users once the signature style has been established the modifications are usually slight. For users whose signatures have changed significantly over time, and such cases do occur although infrequently, the earlier version is almost always completely abandoned and the current version is the only one that is used. Only in some exceptional cases has it been found that a user may recall an old form of his or her signature.

In [63] it is reported that in the experiment with 248 users, three users continually varied between two signatures. This suggests that if a handwritten signature verification system has to verify such exceptional cases of more than one signature by an individual, the system would need to maintain a list of reference signatures over time. Moreover, when a user's signature varies over time, this variation should be taken into account in the design of handwritten signature verification algorithm, assuming that the user might be using elements of a former signature in the current signature. According to [48], in the vast majority of cases the current signature is sufficient for verification purposes.

(4) Automatic Handwritten Text Analysis

There are two main tasks in *automatic handwritten text analysis*—the *handwritten text recognition* and *handwritten signature verification*. The results achieved so far in both areas will be discussed in section 4.3 and 4.4. Several minor application areas connected with handwritten text analysis such as applications in psychology, criminology and various medical and educational applications are not in the scope of the author's research and, therefore, they will not be mentioned below. There are also various approaches to the acquisition of the data to be analyzed. Techniques involved in this process are described in the following section.

4.1 Off-line and On-line Handwriting Analysis

The field of handwriting analysis can be divided into *off-line* and *on-line*:

- Off-line (static) analysis deals with the recognition of text that is written on a paper-like medium. The first processing step of an off-line recognizer is the scanning and digitization of written text. Thus, the input of off-line recognizers only consists of pictorial representations of text.
- In on-line (dynamic) analysis, the data of a written text are recorded during writing. Hence, the additional timing information, i.e. on-line information, of the writing can be utilized to recognize the written text.

On-line information comprises the number of strokes, the order of strokes, the direction of writing for each stroke, the speed of writing within each stroke and pen-up and pen-down information. A *stroke* is usually defined as a sequence of coordinates representing the positions of the pen between a *pen-down* and a *pen-up* movement. A pen-down movement is the action of putting the pen onto the writing surface to start writing. A pen-up movement is the action of lifting the pen from the writing surface to end the writing movement. Some on-line data acquisition devices also measure the pressure of the pen on the writing surface during writing and exploit this information in the recognition process.

Off-line and on-line recognizers require different hardware (Section 4.2). While off-line recognizers only need a scanner to digitize written text, on-line recognizers require a transducer that records the writing as it is written. This implies that words written on paper cannot be used as input for on-line recognizers. In order to use on-line recognition methods, words must be written using special hardware. This restricts the number of practical applications of on-line

recognition techniques. The technology with which tablet digitizers are currently constructed is either electromagnetic/electrostatic or pressure-sensitive. A pressure-sensitive technology has the advantage that it does not require the use of a special stylus. Other approach to acquire data suitable for on-line recognition assumes that the text has to be written by special instrumented pen that enables tracing its location or captures its movements (e.g. by camera).

The temporal information of on-line systems complicates recognition since it records variations that are not apparent in the static images. For example, the letter 'E' can be written by means of various stroke orders or directions. Nevertheless, this can be dealt with successfully and the temporal information provided by on-line entry improves recognition accuracy as compared to off-line recognition [118], [70]. Moreover, the user can alter the way in which he or she writes characters to improve recognition when some of his characters are not recognized properly. In fact, in on-line recognition we not only often encounter adaptation of machine to writer but also adjustment of writer to machine [118], [132].

The different applications of handwriting recognition can be arranged according to the terms on-line and off-line recognition as follows: Process automation, office automation and current applications in banking environments belong to off-line recognition because words are scanned from paper (e.g. letters or forms). For instance, postal automation is a typical example where only written word images are processed by the recognizer. Reading machines for the blind also belong to off-line recognition. *Personal Digital Assistants* (PDA) and methods for signature verification are on-line applications and require special hardware. For instance, pen pressure information used to verify a signature cannot be (easily) derived from the static image but must be measured by special hardware. Nevertheless, some signature verification methods can be applied in the off-line context as well.

4.2 Data Acquisition Devices

A brief overview of data acquisition devices currently used in handwriting analysis will be introduced in this section.

4.2.1 Data Acquisition Devices for Off-line Systems

Traditional table scanners are used in most cases although there are also other types of scanners such as the C-Pen (Fig 4.1).

A C-Pen 800C consists of a digital camera, a processor and memory components making it possible to read and interpret printed text. The digital camera inside the pen captures the text and saves it in C-Pen's memory as a document that can be transferred to a PC, PDA or mobile phone using cable or *infrared* (IR) communication as it supports both serial cable connection (RS232 interface) and infrared communication (IrDA standard). The camera (Fig. 4.2) is a CMOS sensor that captures 50 pictures per second with a resolution of 300 dpi [144].

The standard version of the C-Pen allows the user to collect and save printed text only, although the developer of the pen also offers the *C Write* functionality which allows the user to write characters using the C-Pen like an ordinary

Figure 4.1: C-Pen 800C made by C Technologies

ballpoint pen (except for the fact that there is no ink). The C-Pen can follow its own movement over a surface and recognize the movement as a letter or numeral.[1] If the C-Pen is used in this way it operates as an on-line data acquisition device. Moreover, the developer offers the SDK[2] for public use. Using the SDK it is possible to develop software fitted to any purpose—including signature verification.

Figure 4.2: C-Pen 800C—detailed view

4.2.2 Pen-Based Data Acquisition Devices for On-line Systems

There are several commercial systems for on-line handwriting acquisition available on the market worldwide. Also, there are some under development; nowadays they exist only as prototypes. In this section a survey of all the available input devices will be given.

[1] The principle of movement capture used is the same as is common in optical mouse.
[2] Software Development Kit

I-Pen (Inductum)

The pen works with any paper (no special paper is required), and the main functionalities of the pen are text and drawings capture, mouse-compatibility, direct printout via a laser printer and e-mail. The pen nib on the I-Pen is interchangeable, and the pen can be used as a graphite pencil, ballpoint pen or whiteboard pen. The company applied for a patent for its product in the autumn of 2000 [137].

Figure 4.3: I-Pen, with the lid, the interchangeable nib and the pen body

The I-Pen is based on accelerometer technology, and the pen system consists of a pen lid and a pen itself. When writing on paper, the lid is connected to the paper so it can determine the position of the paper. It is the movement of the pen in relation to the lid that assures a correct reproduction of the notes. If there is no reference point (lid), the pen stores the movements (strokes) in chronological order. The I-Pen is activated when the lid is removed, and the data acquisition starts when the pen nib touches the paper.

Virtual Pen (GOU Lite)

The Virtual Pen or V-Pen works like an ordinary pen and can transmit writing, drawings and email from any surface directly to a mobile phone, PDA or PC. It can also be used as a mouse to control the user's movements around a computer screen, and for digital signatures.

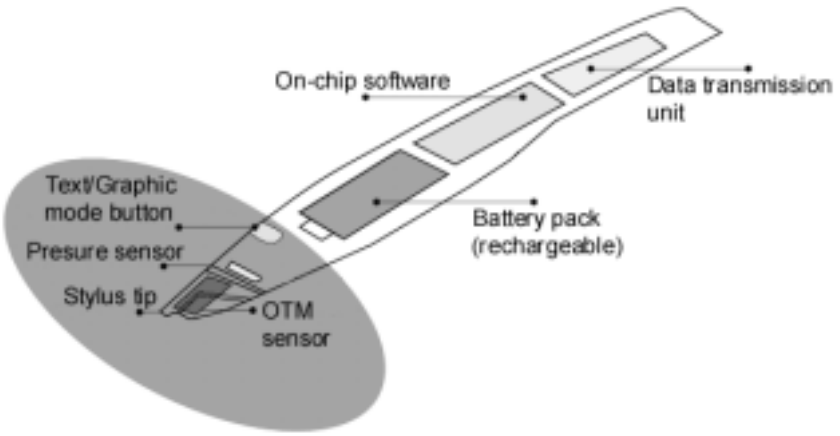

Figure 4.4: V-Pen components

The V-Pen is based on the *Optical Translation Measurement* (OTM), which is a technology that measures the relative movements between an OTM sensor

and an object moving in front of its optical apertures[3]. A digital pen uses the OTM component to measure the movement of the pen nib in relation to the surface, and sends the information using *Bluetooth* technology to a computer.

E-Pen (InMotion)

The E-Pen system captures the user's handwriting and converts it into a digital form. The electronic E-Pen is a patented product containing two components—a receiver module and the pen itself. The pen does not require any special paper in order to work and about 100 USD.

The E-Pen makes it possible for the user to convert handwritten text into a digital form. The receiver module can be connected to a computer, a PDA or a mobile phone

Figure 4.5: E-Pen

The receiver module functions as a measuring device, and it is fastened to the top of the paper on which the user writes. The E-pen uses an external GPS-like[4] system to register and save movements. When writing, the pen transmits ultrasonic waves that are registered by the sensors in the receiver. The receiver contains a microprocessor that processes the information and converts it into digital form, and by connecting the receiver to a computer the information is copied to the computer [138], [139].

N-scribe (Digital Ink)

N-scribe consists of an electronic pen and a pen case, which works as a measuring instrument. Like most digital pens, N-scribe converts handwritten text into a digital code, and the pen does not need any special paper [140].

Figure 4.6: N-Scribe

[3]The same principle is used in optical mouse.
[4]Global Positioning System

N-scribe uses a GPS-like measurement system—the pen case contains a microprocessor that processes the information and converts it into digital codes corresponding to the pen's movement. The information is stored in the pen case as *jpg* images or *pdf* files, and can be sent to a computer via an IR port for further distribution.

Compupen (Pen2Net)

Compupen comes in three different variants: MediPen, SciPen and NotePen.[5] The pens work independently of the surface area and are used in almost the same way as traditional pens. They are equipped with a display, menus and command buttons so the user can supply the pen with information about the context of the text. Without instructions, the pen processes the information in chronological order [141].

Pen2Net states that in a previous phase the pens were based on accelerometer-based technology, but this technology was rejected in favour of an optical technique, which photographs and recognizes different fields of the document after the user has fed the pen with information about the context. It is hard to evaluate this technology because the company fails to explain how recognition of the different fields is possible.

Anoto Technology

The Anoto AB company developed the technology that consists of a digital pen and digital paper. The digital paper is conventional paper with a special Anoto pattern printed on it. When writing on this pattern, the digital pen creates a digital copy of the written information.

The digital pen looks like an ordinary ballpoint pen and is used in the same way. The pen is activated when the cap is removed, and deactivated when the cap is replaced again. It consists mainly of a digital camera, an advanced image-processing unit and a Bluetooth transceiver (Fig. 4.7). The pen also holds a pressure sensor, an ordinary ink cartridge so that the user can see the written information, and a memory that can store several fully written pages.

The pen uses the camera to take digital snapshots of the pattern so the pen can calculate its own position in the entire *Anoto pattern*. These snapshots are taken 50-100 times per second and infrared lights are used to make the dots of the Anoto pattern visible to the digital camera.

The *Anoto pattern* (Fig. 4.8) is printed with carbon-based black ink, and the infrared light interacts with the carbon-based dots. The pattern consists of small dots that are barely visible to the eye; the pattern is perceived as a slightly off-white colour. A small number of dots uniquely define the position in the full pattern[6]. The ink from the pen is not visible to the camera; its only function is to make written text visible to the human eye. This means that it is possible to write on the same piece of the paper over and over again without destroying the digital pattern.

[5]The three different pens from Pen2Net have different areas of application. MediPen is intended to be used by doctors and other hospital personnel for writing prescriptions and medical notes, etc. SciPen can be used by engineers and researchers for report writing and drawings. NotePen is intended for writing such things as study notes.

[6]The full pattern area comprises about 60 million km^2.

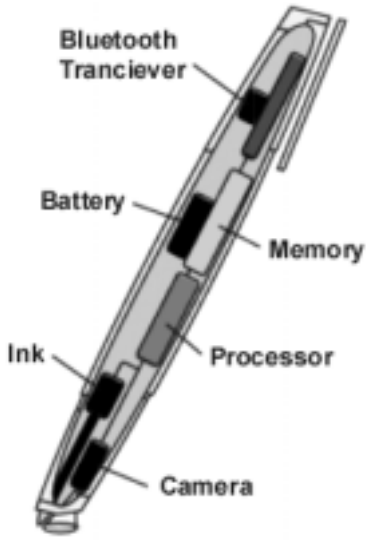

Figure 4.7: The Anoto Pen

Besides calculating the pen's position, the processor is used for gathering and storing information about the angle between pen and paper, the turning of the pen, and the pressure against the paper. For each snapshot an accurate timestamp is collected. All data from the image processor are packed and stored into the memory.

Figure 4.8: The Anoto pattern

The main difference of the Anoto system is the special paper. The digital paper enables to calculate the pen's own position in the entire pattern, this allows its usage for filling in customized forms.[7] On the other hand the necessity of using of Anoto paper is a considerable disadvantage as the paper itself is (i)

[7]Almost any paper can be used together with the Anoto pattern, and the pattern can be

expensive and (ii) the mobility of the pen is limited as user must have the Anoto paper along.

4.2.3 Tablet-Based Data Acquisition Devices for On-line Systems

A large area of the on-line data acquisition device market consists of *tablets* (or *touchscreens*) and many papers report that they are used for on-line data acquisition. The physical device providing the interface between the pen and the tablet is called a *digitizer*. It is the high-resolution hardware that recognizes the motions made with the pen and passes it to the tablet.

The basic purpose of the digitizer in a pen tablet is to translate the position of the pen into x and y coordinate values. There are two basic types of digitizers used today: *active* and *passive*. Both active and passive technologies use similar components to digitize data:

- a pen, stylus or human finger to generate input data
- a sensor device to generate x, y analog coordinates from the input data
- a micro controller to convert the x, y coordinates into digital data
- driver software

In order to better understand the differences between active and passive technologies, a brief overview of how these technologies actually convert analog signal to digital information is given.

Passive Digitizing Technology

Passive technology is used in all PDAs and many vertical tablet applications today. The term *resistive* is used synonymously with the term *passive* in these applications. It's passive because there is no communication between the digitizer and the stylus (pen) and resistive because it is made up of two resistive (conductive) coatings. The digitizers are usually equipped with LCD underneath the digitizer itself in order to visualize the pen strokes, which is more comfortable for the user. The structure of a resistive digitizer is fairly simple. In front of the LCD (if any) there is a sheet of glass that is covered on its top side with a conductive, transparent coating. The coating is made of *Indium Tin Oxide* (ITO). On top of the glass there is a sheet of plastic that is covered on its bottom side with the same conductive coating. The top of the plastic sheet forms the writing surface. In between the glass and the plastic sheet are tiny transparent spacer dots. When user presses down ("*taps*") on the plastic sheet, it contacts the bottom glass and completes an electric circuit via the two conductive coatings. A controller chip measures the resistance from the contact point to each of the four sides of the digitizer and calculates the location of the contact point.

used in catalogues, magazine, calendars, diaries, etc. The paper can be in any desired size or shape depending on its intended use. Any color except carbon-based black ink can be used to print on top of the Anoto pattern without disturbing the function of the digital pen.

During the design phase, a resistive digitizer can be optimized for finger-touch or for pen-touch. This is accomplished by varying the distance between the spacer dots. If the dots are far apart, a broad surface (such as a finger) can depress the top plastic sheet enough to make contact with the bottom glass. If the dots are close together, a smaller surface (such as a pen) is required to depress the top plastic sheet enough to make contact with the bottom glass. Standard digitizers are designed with medium spacing between the dots, which allows either a finger or a pen to be used. The problem with this compromise is that when the user rest hir/her hand on the screen while writing, the crease in user's palm may trigger the digitizer instead of the pen tip. This is called the "palm effect". The digitizers used in a pen tablets for handwritten text recognition and signature verification are designed with close spacing between the dots, which produces excellent "palm rejection"—meaning that the edge of the hand will not accidentally trigger the digitizer. This makes the pen tablet much easier to use, particularly in a large-screen model.

In a standard resistive digitizer, the amount of light emitted by the LCD that gets through the digitizer (called the transmissivity of the digitizer) is between 75% and 83%. In addition, because there are four distinct surfaces (both sides of the plastic sheet and the glass sheet), a significant amount of ambient light (typically 20%) is reflected from the digitizer. The user perceives this reflected light as glare; it also has the effect of reducing the contrast of the LCD image. The state-of-the-art digitizers add one key ingredient that significantly improves all of these problems: a silicon oil-based liquid that replaces the air between the plastic and glass layers. The liquid makes the digitizer seem like a single unit instead of two separate layers. It improves the transmissivity of the digitizer to over 90%, and reduces the reflected light to less than 10% [133].

Other passive digitizing technologies used primarily in non-handwritten analysis applications are described below.

Capacitive Digitizing Technology utilizes a voltage that is applied to the four corners of a screen. Electrodes spread out the voltage creating a uniform voltage field. The touch of a finger draws current from each side in proportion to the distance from the edge. The controller calculates the position of the finger from the current flows. Capacitive technology is used in video games, kiosks and point-of-sale devices.

Near-field Imaging Digitizing Technology is similar to the capacitive technology described above, except that an electrostatic field (rather than an electric one) is used. The touchscreen is the sensor and it can detect a conductive object like a finger or conductive stylus through the glass. The technology is used in industrial applications.

In the *Acoustic Wave Digitizing Technology*, two ultrasonic transducers are mounted on two edges of the display, setting up a pattern of sound waves. When a finger or other energy-absorbing stylus is inserted, it disturbs the pattern. A controller calculates the location of the finger from the changes in the sound. This technology is used mainly in medical monitoring applications and in kiosks.

Infrared Digitizing Technology is similar to the acoustic wave technology described above, except it utilizes infrared light, rather than ultrasonic sound. This technology is used mainly in large displays, banking machines and military applications.

Active Digitizing Technology

An *active digitizer* is one in which the input device (usually a pen) contains some electronics external to the touched surface of the digitizing device.

Active digitizer technology is the technology of choice for applications in which higher resolution, higher accuracy and the ability to *hover* (also called mouseover or rollover) is required. It is also the technology of choice for Microsoft's Tablet PC initiative.

An active digitizer use an active technology utilizing *electromagnetic resonance*. It is called active because it uses *radio frequency* (RF) technology to actively communicate the position of a special stylus to the surface of a sensor grid that usually sits underneath the LCD (if any).The special stylus gives active digitizers the capability of proximity sensing or *hovering*. This means the sensor grid can determine the position of the stylus when it comes within 5–10 mm of the writing surface. When the sensor grid recognizes the presence of the stylus, it responds by moving the cursor directly under the position of the stylus. As long as the stylus is within range of the sensor grid, the cursor follows the movement of the stylus. This action is called *hovering*.

Other active digitizer technology is based on *External Global Positioning System* (External GPS)—the technology for tracking a moving object in 2-D space without a digitizing tablet. This system can track the absolute position of an object, but requires an external system and a predefined working area [136]. The GPS can be implemented with numerous technologies; the simplest one is the *Mechanical Tracking System*. This technology links a pen to a reference point, like a paper clip, via a physical link. The goal is to achieve a 0.3 mm resolution without a digital surface and without making the physical link an obstacle to free and comfortable handwriting. The writing is done with a ballpoint pen and allows the user to write on any paper.

Another way to implement the GPS is with acoustic technology. This technology is based on measuring the time it takes for a sonic impulse to travel from a sound generator (pen nib) to a receiver (microphone), so that the distance can be calculated. By positioning two microphones on the drawing area, the two-dimensional area of the stylus can be calculated (the same principle using three microphones can be used for three-dimensional space). A problem with this method is that echoes of the sound signal can be reflected and cause reception of "ghost" pulses by the receiver.

Digitizer Technology—The Differences

Data acquisition devices for handwritten text recognition and signature verification applications use either resistive (passive) or electromagnetic (active) digitizing technologies. Both the advantages and potential disadvantages of these technologies are discussed below.

Faster data conversion rate — A typical resistive digitizer samples data at a rate of 40 samples per second. Active digitizer sample rates typically exceed 130 sample rates per second. This allows the user to write at their normal speed while capturing all of their input consistently and without a jagged appearance across the surface. Typically a minimum of 100 samples per second is required for good English handwriting recognition.

Resolution — A typical resolution of a resistive digitizer is 100-150 ppi[8] (4-6 points per mm) versus resolution of 1 000 ppi (40 points per mm) in case of an active digitizer. The higher resolution allows data to be captured with higher precision.

Accuracy — The accuracy of an active digitizer is typically ±0.25 mm versus ±2 mm for a passive digitizer. With an active device you can repeatedly touch the digitizer with the pen and get a point returned that is within 0.25 mm of the actual point of contact. This allows the user to produce an accurate electronic representation of his or her data as it was actually drawn or written.

Supports hover capability — Hover refers to the ability of the cursor to track pen movement without clicking. Adding this capability to a resistive digitizer is not easy.

Easier and more stable calibration — Resistive digitizers require a calibration step during manufacture. Active digitizers contain their calibration in the firmware so they can be assembled and used without recalibration. Additionally, resistive digitizers are sensitive to temperature variations and recalibration by the user is occasionally required.

Clarity of screen — Active digitizers use an underlay (behind the LCD) versus resistive digitizers that use an overlay (on top of the screen). The overlay is typically a PET film. The stack up used in a resistive digitizer (usually PET film, two ITO coatings and the substrate) reduce the transmissivity by cca 80%. This makes the screen seem dimmer and dulls the contrast of the screen.

Enhanced durability — The PET top sheet layer used in resistive digitizers is subject to scratches and stretching due to its thermoplastic nature. This can affect the performance of the system and potentially shorten its useful life. In addition, the ITO coating can wear off leaving a dead spot on the digitizer after a repeated number of touches. In an active system all of the critical components are not exposed to or affected by the environment.

Special pen have to be used — With an active system, if the user loses the pen, the system stops working since some of the electronics are located in the pen. With a passive system any sharp object can be used in a pinch, including a fingernail. To overcome this potential disadvantage, the use of a tethered pen is an option.

Price — Active digitizers systems are more expensive then passive ones due to the electronics located in the pen. The value of active technology, as described in the previous section, outweighs this additional cost in many applications where resolution, accuracy and the ability to hover are required.

RF signal transmission technology difficulties — Active digitizers use RF for signal transmission that makes them inherently more complex to implement (noise, shielding, etc.). The primary difference between the active

[8] pixels per inch

digitizer technologies supplied by the two leading suppliers (FinePoint Innovations and Wacom) is in the way in which this potential disadvantage can be minimized.

Thickness — Active digitizers are thicker then passive ones. Typically an active one is 0.4 mm thicker than a passive one. This can be an important consideration in the design of the OEM device.

There are several important industry standards for the tablet to computer interfaces. Wintab drivers provide a standardized set of functions that developers of applications for Microsoft Windows can use. The applications programming interface (API) that the drivers provide is an open industry standard interface. With Wintab drivers, hardware vendors can only provide one standard driver for all software applications supporting the standard digitizer interface and software vendors are assured that their software will work with all input devices from a variety of manufacturers. Wintab drivers support multiple input devices and multiple channels.[9] Therefore, it is possible to use a tablet and a mouse at the same time. All the mainstream commercially available tablets are supplied with Wintab drivers [146].

PenX [147] is an operating system extension for Microsoft Windows which provides a common API for pen and tablet drivers to enable accurate inking, ink capture and handwriting recognition.

iSign [148] is a software development kit (SDK) for implementing Business to Consumer (B2C) and Business to Business (B2B) solutions that support handwritten signature verification over the Internet. It provides real time signature capture, verification and binding, as well as ink display and encryption technologies.

Figure 4.9: Wacom tablets—Graphire2, Intuos2 and Cintiq

There are two commercially successful tablets on the market—both used for on-line data acquisition by many research groups [142], [143]. The most often mentioned tablets in scientific papers are Wacom tablets [149] which utilize batteryless, cordless technology: the low-cost Wacom Graphire2, the high-resolution Wacom Intuos2, and the Wacom Cintiq implementing LCD technology to achieve more natural paper-like experience (Fig. 4.9).

[9] Wintab supports x, y and pressure data, and also cursor tilt information in case that the tablet reports it.

Figure 4.10: ePad-ink tablet

The ePad-ink is a LCD equipped tablet that is compatible with most commercial signature verification systems and can be used with proprietary electronic forms. In addition, it is designed for a use with *point-of-sale* (POS) retail terminals. The visual feedback while signing ensures a natural signature and the document data displayed beneath the signature block adds context to the signing of electronic documents, forms and retail transactions.

Figure 4.11: Tablet PC

The last handwriting acquisition device based on active digitizer technology to be mentioned is the *Tablet PC*[10] [150]. The Tablet PC is a fully functional computer (notebook) running the Microsoft Windows XP Tablet PC Edition operating system which offers various handwriting and speech recognition capabilities, so that the user can create, store, and transmit handwritten notes and voice input. Tablet PCs are developed with a primary focus on highly portable solutions and most of them employ Wacom's Penabled sensor technology.

[10]Personal Computer

Camera-based Data Acquisition

In [79] a unique camera based data acquisition systems are discussed. Authors consider all the previously mentioned systems bulky and complicated to use increasing the complexity of the whole recognition or verification system. Cameras are much smaller and simpler to handle and are becoming ubiquitous in the current computer environment. The feasibility of using a visual interface that can be built with video technology and computer vision techniques in order to capture signatures has been reported.

The visual interface allows the user to write on ordinary paper with a regular pen, providing him or her with a more natural and comfortable environment while interacting with the computer.

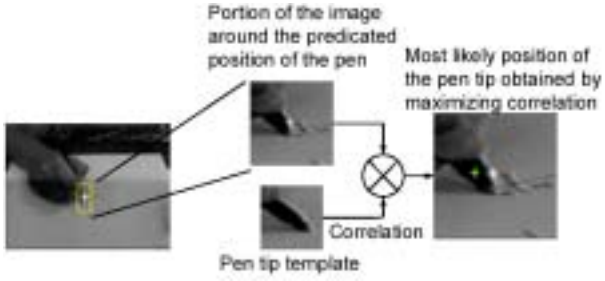

Figure 4.12: Camera-based system: motion sensing principle

The x and y coordinates are obtained from the position of the pen nib in the current frame. For this purpose the *optimal signal detector* is used. The optimal detector is a filter matched to the signal (in this case the segment of the image) and the most likely position of the pen is given by the best match between the signal and the optimal detector.

Assuming that the changes in size and orientation of the pen tip are small between two frames, the most likely position of the pen nib in each frame is given by location of the maximum of the correlation between the kernel and the image neighborhood, as shown in Fig. 4.12.

Using the output of the correlation-based tracker, the filter predicts the position of the pen nib in the next frame based on an estimate of the position, velocity, and acceleration of the pen nib in the current frame. This filter improves the performance of the system since it allows to reduce the size of the neighborhood used to calculate correlation. More details description is given in [80].

Other Data Acquisition Devices

A user identification system using a mouse is proposed in [43] and [116]. The mouse is a low-cost standard device for today's computers which is a significant advantage. However, many graphologists warn that writing with a mouse is very unnatural and therefore the data obtained in this way are unusable for handwriting analysis. Anyway, the results presented in [116] yields 7% ERR and proves that a mouse-written signature can be considered for person identification.

4.3 Handwritten Text Recognition

Hundreds of papers reporting the progress in *handwritten text recognition* (HWR, sometimes also abbreviated as HTR or HWTR) have been published throughout more than thirty years of active research in the domain [33]. Due to improved recognition methods and faster and cheaper CPUs[11], even several commercial products have become available.

There are differences between on-line and off-line recognizers. Most of the handwritten text recognition systems deal with off-line recognition, in which an image is the input. These systems are also referred-to as optical character recognition (OCR) systems. In both on-line and off-line system, the ultimate objective is to convert analog handwritten sentences or phrases (off-line or on-line sources) into a digital form (ASCII[12]). It is worth mentioning that most of the OCR systems available on the market are systems designed for non-handwritten (printed) text and most of them utilize template matching techniques. These systems will not be discussed below as this thesis deals only with handwriting.

The major part of published works are devote to Roman character (Latin alphabet) recognition. Within this framework two main types of handwriting can be distinguished: handprinted words and cursive written words (Section 3.3). There are also other categories that are usually distinguished from the two mentioned above: signatures, kanji[13] [151], [155], arabic [14], [67] and Graffiti alphabet [154].

The handwritten text recognition problem is usually divided into *isolated character recognition* and *word recognition*. Isolated character recognition is considered the easier problem by many researchers as it does not require word segmentation in order to extract characters to be recognized. On the other hand, there are only a few applications that utilize isolated character recognition methods—Graffiti-like (Section 4.3.3) character recognizers used in handheld computers and handprinted words recognizers. In some applications a character recognition algorithm is a part of word recognition system. *Dynamic time warping* (DTW) and neural networks are most often used for isolated character recognition.

Word recognition is a much more active research area. The principles of word recognition are described in the following paragraphs. A straightforward approach to the recognition of handwritten words is to segment the words into single characters and then to classify these segments. This works well with good quality input only but not with cursive script, which is common in real applications. As can be seen in Fig. 4.13, it is difficult to find the correct segmentation: The characters 'o' and 'g', for instance, are almost overlapping. Sometimes it is even impossible to segment discrete characters because of the inaccurate writing, especially on word endings, where sloppy writing often leads to the omission of characters. It is, therefore, desirable to avoid explicit segmentation.

[11] CPU—Central Processor Unit

[12] ASCII—American Standard Code for Information Interchange

[13] Kanji constitute a part of the writing system used to represent the Japanese language in written, printed and displayed form. The term is also used for the collection of all kanji letters. One of the set of glyphs common to Japanese (where they are called *kanji*), Chinese (where they are called *hanzi*), and Korean (where they are called *hanja*).

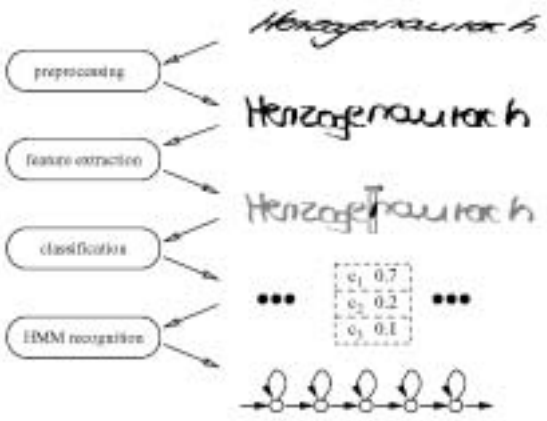

Figure 4.13: Four steps of cursive script recognition

4.3.1 HMM Based Recognition

Most of the current recognizers use *Hidden Markov Models* (HMM) to solve the segmentation problem. An overview of the processing steps for off-line cursive script recognition employing HMMs is given in Fig. 4.13 and the whole recognition process is explained below. Variations to the basic approach are discussed where applicable.

In the off-line systems the aim of preprocessing is to clean the image and to remove all information irrelevant to recognition. Usually stroke width, script size, position, slant and orientation are normalized. Moreover, image has to be transformed from grey level to a bi-level image by thresholding. Good survey of binarization methods can be found in [122]. The main difference between these approached is whether a global or a local threshold is computed. In order to achieve size normalization the image is divided into three vertical zones, namely upper, middle and lower zone. These are limited by four lines: upper line, upper baseline, lower baseline and lower line. These lines are usually detected with simple horizontal projection. Another kind of distortion is slant. There are several slant normalization algorithms known in the literature [12]. However a simple algorithm which checks for maxima in the projection histograms in different slant angles provides good result in most applications.

Preprocessing applied to input data in on-line systems has several additional problems that have to be handled. In almost every on-line system recognizer it is necessary to *compute equidistant points*. In general, the points captured during writing are equidistant in time but not in space. Hence, the number of captured points varies depending on the velocity of writing and the hardware used. To normalize the number of points, the sequence of captured points is replaced with a sequence of points having the same spatial distance. This distance is usually some fraction of the corpus height. If the distance between two neighboring points exceeds a certain threshold, the trajectory interpolation between both points is computed using a Bezier curve [1]. To remove jitter from the handwritten text, every point in the trajectory is replaced by the mean value of its neighbors [50]. *Delayed strokes*; e.g., the crossing of a 't' or the dot of an 'i', are well-known problem in on-line handwriting recognition. These

strokes introduce additional temporal variation and complicate on-line recognition because the writing order of delayed strokes is not fixed and varies between different writers. This stands in contrast to off-line recognition since delayed strokes do not occur in static images. It is one of the reasons why there have been approaches in the recent past trying to exploit off-line data in order to improve on-line recognition rates [49]. Many on-line recognizers apply a simple technique to cope with delayed strokes: They use heuristics for detecting such strokes and remove them before proceeding with feature computation and recognition. Delayd strokes can be easily identified as a short sequence written in the upper region of the writing area, above already written parts of a word, and accompanied by a pen movement to the left.

HMMs model one-dimensional stochastic processes so they can be readily applied to recognition of on-line handwriting, which is recorded as a sequence of sensor data over time. In off-line handwriting recognition, the input is an image which must be transformed into a sequence of features vectors to be processed by linear HMMs. The dimension reduction is not trivial and there are several approaches to achieve one-dimensionality. In [11], a canonical sequence of traversing the edges of the script is computed. Features are heuristics like holes and loops. Others try to segment the input into windows with fixed or varying length, which is slided from left to right along the input image. There are variety of possibilities for feature calculation within this window, such as vertical, diagonal, and horizontal strokes, crossing lines, cusps, loops, i-dots and holes. Other features include also moments [16] and fourier coefficients [31]. As for on-line handwriting recognition neither a standard method for computing features nor a widely accepted feature set currently exists [50]. Common features include vertical position of pen nib relative to baseline, writing direction, and set of features describing the vicinity of pen nib (curvature, pen-up/pen-down[14], aspect of the trajectory, curliness, linearity, slope, ascender/descenders and context maps). Detailed description of these features can be found in [50]. It also should be mentioned the work [49] that addresses the problem of recovering the dynamic information from the static—handwritten word images.

HMMs are stochastic automata where the output is result of two combined random processes: The first process models the structure of the input and is described the probability of state transitions. The second process models the shape variabilities and is characterized by output distributions of a fixed symbol set when entering a model state. The author desist from a detailed description of HMM theory as there are a lot of high quality publications already e.g. [100], [60].

The selection of appropriate model topologies is an important step in designing an HMM recognition system. Although an off-line word image is two-dimensional, most of the literature uses linear models known from speech recognition [31], [39], [53]. When choosing a linear model to represent a written word, it is not clear, how many different states per word should be used and whether the states can be shared with other words. Basically, the following have to be considered: (i) the models must be *trainable*, i.e. there must be enough examples in the training set to estimate the parameters and (ii) the models must be *specific*, i.e. they should represent reality as close as possible. Both aspects

[14] Electromagnetic systems are able to return approximate planar coordinates while pen is in the air, whereas pressure-sensitive technologies cannot.

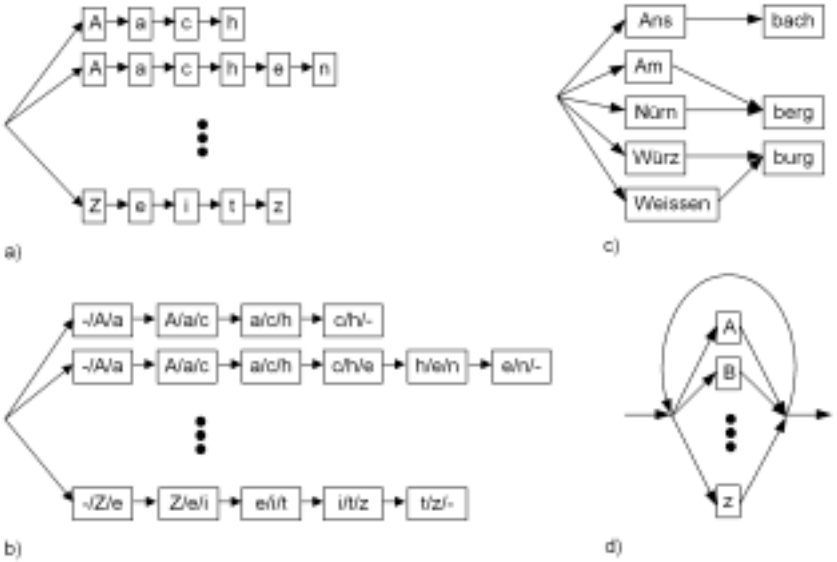

Figure 4.14: Model topologies: a) whole word models, b) whole word models with context depending units, c) context freezing with subword models d) single character models

usually exclude each other: If the models are too specific, then there will be not enough examples in the training set. If they are general, recognition performance decreases because of the lack of discriminating power of the models. The compromise is usually to select some basic models (characters or parts of characters) which are shared across the system and from which the larger models are built. A number of 3 to 5 states per character model has so far turned out to be optimal [53], [113] depending on the complexity of the character to be modeled.

The simple character modelling neglects the influence of context. There are several approaches to context modeling. Explicit models for ligatures are proposed in [18]. Another way to incorporate context is to use different characters models for different contexts. For example: “of” and “if” have different ligatures and therefore two models of ‘f’ are needed denoted by “o/f/-” and “i/f/-”. Like the *triphone* in speech recognition, which is a model of a phoneme in the context of its right and left neighbor phoneme we can introduce the *trigraph* which denotes the model of a character with its two adjacent characters. While good results are reported for on-line recognition [26] and for context modeling on the sub-character level [34], character level context modeling only improves performance slightly [112]. Last but not least context can be modeled by *context freezing units*. To do this it is necessary to look for common subword units in the training set which appear frequently enough to reliable estimate the model parameters. In case of German city names [105] there are common endings like -berg, -burg or -ingen which can be chosen as subword units. Incorporating the freezing units into recognizer improves the recognition rate as such endings are often sloppily written and can only be recognized in context (even for human).

For every unit to be recognized (words, syllables, characters) a separate

model is needed. The recognition result is the model which emits the sequence of symbols with the highest probability. Doing word recognition with HMMs there are several possibilities: The basic recognition model is depicted in Fig. 4.14a. A separate model is supplied for every word in the dictionary, the character submodels are shared between the word models. Fig. 4.14b shows an extension of this approach with explicit context modeling as described in the previous paragraph. A disadvantage of whole word modeling is that a lot of states has to be searched at the beginning of the word (size of dictionary), which is too time consuming. Solution to this is to restrict the dictionary size before recognition [54] or to arrange the dictionary in a tree-like manner [113]. Subword models (as depicted in Figure 4.14c) are somewhat in between single character and whole character models. They have fewer states at the beginning but they need a postprocessing step as the recognized word might not be a valid combination. The last possibility is the looped single character model as depicted in Fig. 4.14d. The advantage of this approach is that we only have a few states at the beginning (size of the alphabet) of the word, which lead to a fast recognition. The disadvantage is the necessity of a lexical postprocessing step: the recognized word might not be in the dictionary and it is necessary to search for best matching word string. Moreover, contextual dependencies can only be incorporated with statistical methods like *bigram* or *trigram* probabilities. Word recognition performance of looped single character models is often worse than with other models, but they can be used to recognize unknown words, which are not (yet) in the dictionary.

Parameter training for HMMs is usually done with *maximum likelihood* method (ML estimation) exploiting training samples. Given the training sample the parameters of the models are estimated in such a way, that the training sample is produced by the model with maximum probability. Clearly, the training sample has to be representative t achieve good recognition results. Unfortunately, there exists no closed form solution for ML estimation of the parameters. Therefore an iterative method is used to get at least a local optimum—the *Baum-Welch algorithm*.

4.3.2 Neural Network Based Recognition

Some researchers prefer to use neural networks to classify characters or parts of characters (*character subimages*). This approach requires the segmentation step as mentioned above. Most authors uses the simple *Multi-Layer Perceptron* (MLP) for characters classification, although other types of neural networks are also reported [9]. A Viterbi-like algorithm can be used to assemble the individually classified characters into optimal interpretation of an input, taking into account both the quality of the overall segmentation and the degree to which each character or character subimage matches the character model. Such system uses two different statistical language models, one based on a phrase dictionary and the other based on a simple word grammar. Hypothesis from recognition based on each language model are integrated using a decision tree classifier [8].

In [50] a *Multi-State Time Delay Neural Network* (MS-TDNN) presented. A MS-TDNN is a recognizer that integrates recognition and segmentation into a single network architecture. This approach was originally proposed for continuous speech recognition tasks [47], [71]. The Multi-State Time Delay Neural

Network is an extension of *Time Delay Neural Network* (TDNN) [123], which has been applied successfully to on-line *single* character recognition tasks. In the experiments reported in [50], each character is modeled using three states representing the first, middle and last part of the character. The MS-TDNN is supplemented with a tree-based search engine [72], which combines a tree representation of the dictionary and with efficient pruning techniques to reduce the search space without losing much recognition performance compared to a flat exhaustive search through all words in the dictionary.

4.3.3 Gesture Recognition

Autonomous research area is gesture recognition (a simplification of handwriting and full gesture interpretation) which has been implemented in handheld computers (PDAs)[15] [154] and at present it starts to be widely used in desktop computers as well. The first commercially successful recognizer was introduced in Palm Operating System (Palm OS) by Palm Computing Inc. in 1992. The Palm OS uses as an input the handwritten pen strokes that are written directly on the screen. These strokes are called Graffiti (Fig. 4.15). Because of the patent protection it is not known, how the Graffiti recognition algorithm works. On the other hand there is a number of Graffiti-like recognizers available [156], [157], [158] that use the same principles for recognition and whose recognition rates are similar to Graffiti.

Figure 4.15: Example of Graffiti letters

The recognition algorithm processes a stream of coordinate pairs that describe the ordered path that the pointing device followed when the stroke was entered. These coordinate pairs are assigned to "bins"—regions of the stroke defined by a three by three matrix superimposed over the region of space in which the stroke travels.

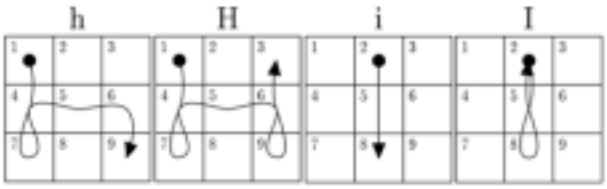

Figure 4.16: Example of XScribble gestures for characters h, H, i and I

The upper and lower bounds of the matrix are defined by the maximum and minimum vertical coordinates in the stroke sequence. The left and right bounds of the matrix are likewise determined. The special case of an extremely tall or an extremely wide stroke needs to be given further treatment. If such

[15] PDA—Personal Digital Assistant

an unbalanced stroke is encountered, it is most likely an attempt at drawing a straight line that has some error in the perpendicular dimension caused by imperfect control of the pointer by the user. Therefore, if the maximum horizontal delta between any two points in the stroke is more than four times the maximum vertical delta of any other two points in the stroke, (the stroke is wide and short), then the dimension of the horizontal side of the matrix is also used for the vertical dimension. This operation is likewise considered for the case when the stroke is tall and narrow.

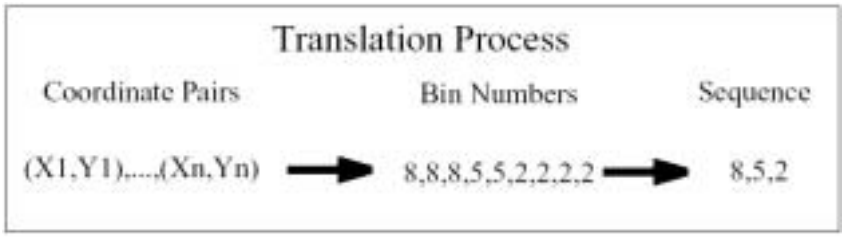

Figure 4.17: Steps of translation process

Once the matrix has been defined, each coordinate pair is assigned a bin number according to its position in the matrix. This sequence of bin locations is then processed with a low-pass filter to remove any short excursions into other bins which have an insignificant number of coordinate pairs falling into them. Then, multiple adjoining occurrences of bin numbers are compressed into single digits. In this way, the sequence of bins that the stroke has passed through is determined and corresponding character (or command) is chosen. Figure 4.17 shows the flow of data from beginning to end.

4.3.4 Databases of Handwriting Texts

In order to compare the accuracy of handwriting recognition systems, several databases were created in last few years. The most often cited are the Unipen and the NIST databases.

Contrary to other pattern recognition fields, such as speech recognition and optical character recognition, until 1999 no significant progress was made in on-line handwriting recognition to make large corpora of training and test data publicly available, and no open competitions have been organized. In 1999 the international Unipen Foundation installed a large database of on-line handwritten samples, collected by a consortium of 40 companies and institutes donating over 5 million characters, from more than 2 200 writers. The data can be obtained on CD-ROM [153].

The Unipen data format is gaining in popularity. Even institutions not involved in the official Unipen benchmarking process use the Unipen data format for on-line handwriting data. Although not particularly condensed, the format is in legible ASCII and very flexible.

During the year 2001, new initiatives have been developed, notably in the area of an XML-based[16] format for on-line handwritten data which has properties suitable for applications in pen-based computers. It is the intention of

[16] XML—eXtensible Markup Language

the involved industrial group to integrate these efforts within the MPEG7[17] standard.

NIST[18] offers several databases of handwritten characters frequently used to test off-line handwritten recognition systems [152]: The SD19 database is composed of more than 800 000 hand printed characters written by 3 600 different writers. Characters are already segmented and are presented in images 128×128 pixels. The SD11 database contains images of 13 500 miniforms and files containing ASCII transcriptions of the strings that were written in the miniform fields.

4.4 Handwritten Signature Verification

Signature verification refers to a specific class of automatic handwriting processing: the comparison of a test signature with one or more reference specimens that are collected as a user enrolls in the system. It requires the extraction of writer-specific information from the signature signal or image irrespective of its content. This information has to be almost time-invariant and effectively discriminant.

The signature verification problem has been a challenge for about three decades. Three survey papers [94], [92], [96] and the special issue of the journal [59] have summarized the latest developments in this field. In this section a brief overview will be given by focusing on the major works by the various teams involved in the domain.

4.4.1 Evaluation of Signature Verification Algorithms

Signature verification attempts mainly to exploit the singular, exclusive, and personal character of the writing. In fact, signature verification presents a double challenge. The first is to verify that what has been signed corresponds to the unique characteristic of an individual, without having to worry about what was written. A failure in this context, i.e., the rejection of an authentic signature, is referred to as a type I error (see Section 2.1). The second challenge is more demanding than the first and consists of avoiding the acceptance of forgeries as being authentic (type II error). The tolerance levels for applications in which signature verification is required is smaller than what can be tolerated for handwriting recognition for both type I and type II errors, In some application, a bank, for example, might require (unrealistically) error of 1 over 100 000 trials for type I error [38] and even less for the type II error. Current systems are still several orders of magnitude away from these thresholds. System designers have also had to deal with the trade-offs between type I and type II errors and the intrinsic difficulty of evaluating and comparing different approaches. Actually, the majority of the signature verification systems work with the an error margin of about 2–5% shared between the two errors. All reduction of one error inevitably increases the other (Section 2.1).

The evaluation of signature verifications algorithms, as for many pattern recognition problems, raises several difficulties, making any objective comparison between different methods rather delicate, and in many cases, impossible.

[17] MPEG—Moving Picture Expert Group
[18] National Institute of Standards and Technology

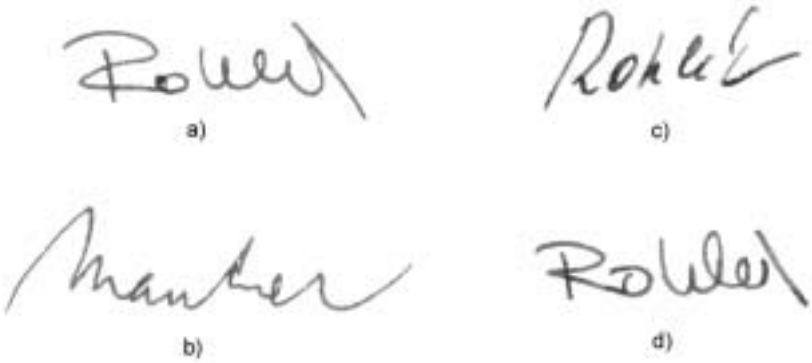

Figure 4.18: Types of signature forgeries: a) genuine signature, b) random forgery, c) simple forgery, d) skilled forgery

Moreover, signature verification poses a serious difficulty, which is the problem of type II error evaluation, or the real risk of accepting forgeries. From a theoretical point of view, it is not possible to measure type II error, since there is no mean by which to define a good forger and to prove his or her existence, or even worse, his or her nonexistence. However, from a practical point of view, several methods of type II error estimation have been proposed in the literature, i.e., that is picked up on a random basis the true signature of a person and considering it as a forgery of the signature of another person (*random forgery*, also called *zero-effort forgery*). Many studies incorporate *unskilled* (*simple*) *forgeries*, which is represented by a signature sample that consists on the same characters as the genuine signature, although the forger has no knowledge of the genuine signature shape (he or she only knows the name). In some rare cases, highly *skilled forgeries* are used which are suitable imitations of the genuine signature as the forger has the genuine signature available while writing the forgeries. A skilled forgery has almost the same shape of the genuine signature, therefore it is more difficult to detect (Fig. 4.18).

The definition of all this terminology, random, unskilled, and skilled forgeries, are rather discretionary and vary enormously from one benchmark to another, making the evaluation of this type of error extremely vague and certainly underestimated [94]. As in many other shape recognition domains, it is very difficult to compare the results of different systems.

One way this can be done is to compare different systems by taking several of the systems available on the market and testing them under the same conditions as the two systems that are about to compare [78]. That would be a laborious task to test ten or so systems (cost of operation, mobilization of equipment and personnel, etc.), but certainly be feasible, and very useful in making a final decision between two or three prototypes.

Another approach is to use a public-domain database [128], [129] and determine the error rates on the test group of signatures.

Unfortunately, there are algorithms that cannot be compared by those methods. These are algorithms that are designed especially to some specific data, that are usually produced be special input devices, and therefore are incomparable to data available in public-domain databases.

From the practical point of view only the skilled signatures should be used to test signature verification systems, although most of the scientific papers presents results of tests made using random forgeries. On the other hand, genuine signatures are not easy to forge and laboratory tests generally take into consideration poor forgeries obtaining, of course, better experimental results.

Building a test database of signatures that is representative of realworld applications is quite a difficult task since it is difficult enough to find people that will willingly sign 10 or 20 times. People are not always happy to have their signatures stored in a computer or given to others to practice forging them. Therefore most test databases are built using signatures from volunteers from the research laboratory where the signature verification research has been carried out and as a result most test databases have very few signatures from people that are old, disabled or suffering from a common disease (for example arthritis). Percentages of such people in the population is significant and these are the people whose signatures are likely to pose the greatest challenge for signature verification. Moreover, there is a great deal of variability in signatures according to country, time, habits, psychological or mental state, and physical and practical situations [94]. It has been reported that FAR and FRR are generally higher when the systems are used by a more representative group. The higher FRR by a more representative group is not surprising since the signing environment generally is not as consistent as it often is when signatures for a test database are collected.

Most researchers have their own test signature databases with a varying number of genuine signatures, some have skilled forgeries while others do not, some have screened the signature database to remove some signatures that for some reason were not acceptable while others have done no screening, the number of signatures used in building a reference signature often varies, different tests and thresholds have been used and even different definitions of FAR and FRR have been used. Some studies use a different threshold for each individual while others use the same threshold for all individuals. This is a rather sad state of the art in the handwritten signature verification evaluation.

It should be noted that the aims of authentication are going to be different for different types of applications. For example, the primary concern of verification in a credit card environment (where the card holder presents a card to make a purchase and signs on an electronic device that automatically verifies the signature) must be to have zero or near zero false rejection rate so the genuine customer is not annoyed by unnecessary rejections. In this environment fast verification is essential and, in addition, the information required for signature verification should not require too much storage since it may need to be stored on a credit card strip or a smart card memory. A high level of security against forgeries may not be required and a false a FAR of 10% or even 20% might be acceptable since even that is likely to assist in reducing credit card fraud as that would be much better than the minimal checking that is done currently. On the other hand, in a security sensitive environment that was, for example, using handwritten signature verification for granting an authenticated user access to sensitive information or other valuable resources, it would be necessary to have a high level of security against intruders and a zero or near zero FAR. A FRR of 10% or higher would be a nuisance but might be acceptable. Of course an ideal signature verification system should have both the FRR and the FAR close to zero but no technique of signature verification presently appears capable of

performing consistently at this high level. It should be noted that FRR and FAR are closely related and an attempt to reduce one invariably increases the other (Section 2.2.2).

Technique which promises a small FRR when tested on an entire database does not guarantee a small FRR for each individual. The performance figures reported in the literature are normally aggregate figures and it is not uncommon to find some individuals that have much larger error rates than the rest of the population in the test database. Of course, it is desirable that a technique not only have good aggregate performance but also good individual performance.

4.4.2 Overview of Signature Verification System

The design of a signature verification system requires solutions of five types of problems (see Figure 4.19):

- Data acquisition
- Preprocessing
- Feature extraction
- Comparison process
- Accept/reject decision

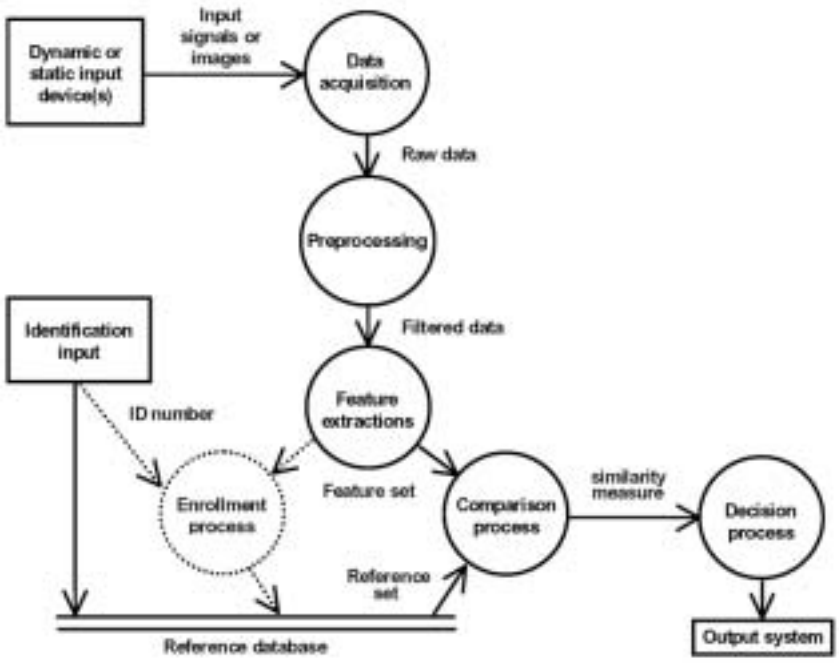

Figure 4.19: Data flow diagram of signature verification system

Data Acquisition

An off-line signature verification system receives an image as input from a camera or scanner. An on-line signature verification system gets its input from an electronic pen, digitizer or from other input device (e.g. camera, mouse). The signature is then represented as one or several time-varying signals (Section 4.1).

Preprocessing

In on-line systems *noise reduction* and *length normalization* is accomplished by traditional techniques. Noise reduction is usually carried out by *Fourier analysis*. Length normalization can easily be performed by several different approaches, among the others a simple approach is based on a *B-spline interpolation* procedure and on a fast *resampling* process [90].

A more complex task is the segmentation of the signature into basic strokes. Some consider a segment as each piece of written trace included between pen-down and pen-up movement of the pen and call it component or fundamental stroke [23]. Others use curvilinear and angular velocity signals of the pen movements to locate *components* and *strings* [98]. A more recent segmentation technique is based on a *dynamic splitting* procedure whose basic idea is to perform the splitting using the information about both the reference signature and the input signature, in fact the reference signature [25].

An off-line signature verification system must be able to process a signature apposed on specific paper forms (like checks). In this kind of systems the preprocessing phase deals with *localization* of the signature in the picture, the *extraction* of the signature from the background, the *thresholding*, and the *filtering*. While noise reduction can be performed by Fourier analysis, the extraction of the signature image from the background is usually solved with the help of *window operator* [90]. A more complex task is the segmentation as no dynamic information is available [49].

Feature Extraction

The primary issue in handwritten signature verification is of course "What aspects or features of a signature are important?". There is no simple answer to this but two different approaches are common.

In the first approach, all of the collected signal values of a signature are assumed important and the test and reference signatures are compared *point-to-point* using one or more sets of these values. In this approach the major issue that arises is how the comparison is to be carried out. Perhaps the signatures could be compared by computing the correlation coefficient between the test signature values and the corresponding reference signature values but point-to-point comparison does not work well since some portions of any two genuine signatures of the same person can vary significantly and the correlation may be seriously affected by translation, rotation or scaling of the signature. Another approach might be to segment the signatures being compared and then compare the corresponding segments using some alignment of segments if necessary. This approach works somewhat better and is discussed in more detail later.

In the second approach, all the available values are not used. Instead, a collection of values are computed and compared. These are called *features* and some examples that have been used in the studies that are discussed below are:

- Total time taken in writing the signature.
- Signature path length: displacement in the x and y directions and the total displacement.
- Path tangent angles: profile of their variation and average or root mean square (RMS) values.
- Signature velocity: profiles of variations in horizontal, vertical and total velocities as well as their average or RMS values.
- Signature accelerations: variations in horizontal and vertical accelerations, centripetal accelerations, tangential accelerations, total accelerations, as well as their average or RMS values.
- Penup time: total penup time or the ratio of penup time to total time.
- Coefficients of the Fourier, Walsh, Haar or Wavelet transform
- Slant and coefficients derived from Hadamar transform

The above list is far from comprehensive. For example, in [21] 44 features are proposed that include some of the features listed above as well as several others. In a United States Patent [89] propose more than 90 features for consideration.

There are also two approaches to the problem of feature selection: global and local. Global approaches analyze the signature as a whole, local approaches obtain features concerning with segment of the signature. Local approaches are usually more time-consuming than global ones but they allow more accurate verification [24].

Once a set of features has been selected, there may be no need to store the reference signature and only the features' values of the reference signature need be stored. Also, when a test signature is presented, only the features' values are needed, not the signature. This often saves on storage (storage may be at premium if, for example, the reference signature needs to be stored on a card) and that is why representing a signature by a set of values of its features is sometimes called *compression* of the signature.

Comparison

On-line signature verification methods can be classified in two main groups. The first group contains methods dealing with functions as features. In this case, the whole signals (usually position, velocity, acceleration, pressure vs. time, etc.) are considered as, or represented by, mathematical functions whose coefficients directly constitute the feature set. In the second group, the methods refer to parameters of the signal as features (total time, means, number of zero crossings, etc.) which are computed from the measured signals. Most of the on-line verifiers published are feature-based while structural verification systems are rare.

Assume that the reference signature is based on a set of sample signatures and for each element of the set of selected features the mean and standard deviation of the feature values have been computed. Therefore the reference signature consists of two vectors: a vector (R) of the means of the features' values of the sample signatures and a vector (S) of the standard deviations.

Clearly to obtain good estimates of the mean and the standard deviations of the features' values of the genuine signatures population it is necessary to have several sample signatures. A larger set of sample signatures is likely to lead to a better reference signature and therefore better results but it is recognized that this is not always possible. The distance between a test signature and the corresponding reference signature may be computed in several different ways. The following six approaches are usually considered [61]:

- Linear discriminant function
- Euclidean distance classifier
- Dynamic programming matching technique
- Majority classifier
- Markov models and hidden Markov models
- Neural networks

Linear discriminant function is a linear combination of the components of feature vector x, has the general form: $G(x) = w^T x + w_0$ where w is a weighting vector and w_0 a constant. w_0, also called the *threshold weighting*, specifies the boundary between two classes—between genuine signature and forgery.

It should be noted here that algorithms such as the gradient descent algorithms which require, in general, a large training set are precluded by the fact that the reference set is generally small. Two particular approaches to linear classification are proposed in [61]. The first has each feature value t_i of the test signature (T) normalized by the reference mean r_i; the second approach has feature value t_i normalized by the reference standard deviation s_i.

Euclidean distance classifier. The Euclidean distance discriminant function is used quite widely [21], [84]. The Euclidean distance metric has the following form:

$$G(T) = (1/n) \sum_{i=1}^{n} (\frac{t_i - r_i}{s_i})^2$$

where, as defined earlier, T is the test signature and r_i and s_i are, respectively, the i^{th} feature's reference mean and reference standard deviation.

Dynamic time warping. Put very simply dynamic time warping (DTW) involves minimizing the residual error between two functions by finding a warping function to rescale one of the original functions time axis. Distance measure between two functions is proportional to the number and types of warping operation performed. DTW is further explained by [61] and is investigated in a number of papers in the literature, e.g. [88].

Majority classifier. The main drawback of the linear classifier and Euclidean classifier is that the FAR tends to 100% as the FRR approaches zero. This is because of the fact that any single feature unduly influences the decision result when deviating far from the mean value, even if the other features have values close to their means for the genuine reference set. One way of alleviating this problem is to use the socalled majority classifier which, is based on the "majority rules" principle. That is, it declares the signature being tested to be genuine if the number of feature values which pass a predetermined test is larger than half the total number of tested features.

Markov models and hidden Markov models. Markov models (MMs) are often used to describe the chains of events in which every event corresponds to a clearly observable state in the model [135]. The model can be trained to produce a sequence of frames corresponding to a genuine signature with a high probability. The distance between the test signature and the reference signature can be expressed as the probability that the sequence of frames corresponding to the test signature is (not) produced by Markov model trained by reference signature(s). Hidden Markov models (HMMs) can be used in a similar way [101], [42].

Neural networks. Neural networks are used in pattern recognition to classify or cluster objects represented by feature vectors. Various neural networks are reported to be used for signature verification [75], [44], [29]. As different neural networks require different input data and different training algorithms, the author desist from a detailed description of various types of neural networks and provides necessary details where needed (Section 4.4.3 and Section 5.2.2). A good survey of neural networks is in [130], [62], [57], [35], [13].

On above mentioned approaches any of the following matching strategies can be applied [24]:

- The simplest strategy is based on a *holistic approach* where the test signature is matched with each one of the N reference signatures that are considered as a whole. This approach does not allow any regional evaluation of the signature.

- Another strategy is based on a *regional matching* approach. In this case the test signature is split into segments as well as each reference signature. The matching between the test signature and each of the reference signatures is performed by matching the correspondence segments. This approach allows a regional analysis of the signature but it is carried out in one-by-one comparison process. Therefore a test signature is judged to be genuine specimen if and only if a reference signature exists which is similar to the test signature in each one of its split regions. This matching approach requires quite a large set of reference signatures.

- The most powerful matching approach is the *multiple regional matching*. In this case each segment of the test signature is matched against the entire set of the corresponding segments in the set of reference signatures. Therefore for each segment of the test signature a verification response can be obtained. Then the rest signature is judged to be genuine specimen if each segment is judged to be genuine. This approach allows a regional evaluation of the signature without requiring a large set of reference signatures.

Decision Process

Important problem that strongly affect the effectiveness of the accsept/reject decision is the selection of the optimal set of reference signatures and the selection of the optimal set of personal thresholds. In some cases the procedure for the selection of reference signatures considers both genuine signatures and forgeries. This approach is weak since in many practical cases fraudulent specimens

are not available or they are of poor quality. Then it is much more significant to select the reference signatures directly from the set of genuine specimens and without any information about forgeries.

Personal threshold are usually obtained by comparing the reference signatures among themselves, taken two-by-two and storing the worst or mean result of comparison [98].

The verification process can follow two-level strategy. For example, at the first level, the segmentation results are used to perform a fast rejection of poor forgeries. At the second level, each stroke of the test signature is matched against each corresponding stroke of the reference signatures and the overall result is used to judge the whole signature [24], [25], [83]. Even three-level strategy have been published [93].

4.4.3 Review of Earlier Work

Given the importance of handwritten signature verification, the volume of published literature in the field is not large. This is primarily due to the high perceived commercial value of innovations in the field and perhaps the reluctance of industry to make public the results of their research and development. Most companies that are carrying out research in this field are keen to protect their inventions by keeping the research confidential within the company or by patenting their innovations; in either case much of the work does not get published in the learned journals. This review will include some of the patents in the field of handwritten signature verification.

Some of the early work in handwritten signature verification is not easily available but has been cited in [46] and [64]. The earliest cited work on handwritten signature verification appears to be [73] (1965) that used power spectral density and zero-crossing features extracted from pen acceleration waveform measurements.

The very first signature verification method that will be introduced here is quite straightforward but also quite common.

Description of a simple dynamic signature verification technique based on features that are easy to determine and compute is given in [40]: total time, number of sign changes in the x and y velocities and x and y accelerations, number of zero values in the x and y accelerations, pen-up time, and total length of pen nib path.

The digitizer used in their experiments is a graphics tablet able to capture the signature as samples of (x, y) coordinate pairs at 200 Hz. With such equipment, it is straightforward to compute velocities and accelerations from the data. Smoothing the data, i.e. averaging out the measurement errors, is done to obtain better approximations to the velocities and accelerations. Authors reports that removing dropouts and peaks is sufficient.

For the comparison process a reference is needed—the mean (R) and standard deviations (S) of the values of the features of 5 to 10 sample signatures are computed. To verify the test signature (T), the distance vector (D) is computed $D = R - T$, and normalized by dividing each value by the corresponding standard deviation in the vector S to obtain a vector Z whose norm is then computed. In practice, it is possible to have standard deviations of zero. In

such case, a value or 10% of the mean value is used for the standard deviation in order to avoid division by zero.

The computed norm is then compared to a pre-defined threshold and the signature is authenticated only if the norm is smaller than the threshold. The value of the threshold depends on the application. Best results reported on experiments with random forgeries are 2.5% FRR and 8.6% FAR. Despite the fact that the results are not very good the fundamental problem of signature verification can be observed here—the difficulty of the threshold setup[19].

Another approach to signature verification is to employ neural networks. There are several neural network topologies that can be used, as well as several learning algorithms.

The use of multi-layer perceptron is reported e.g. [29], [85]. After data acquisition and preprocessing of the input signal, the neural network measures the likeness to the various trained patterns. A final decision-making step is still required to qualify the most likely result from all others. Depending on the application, these results are threshold to minimize the FRR and/or FAR. In other words, the nature of the application differs only in this final stage if compared with the simple method mentioned above. The neural networks proposed in [29], [85] differs from other neural networks because they has only one output neuron. Test signature is accepted as genuine if the output value of the neuron is above threshold level (which is set manually).

Multi-layer perceptron based verifier of Japanese signatures has also been used in [120] and yields 3% EER although it has not been reported whether random or skilled forgeries had been used. An instrumented pen that measures the pressure on the paper has been used for data acquisition and simple preprocessing method were applied to raw data—low pass sum filter, amplitude normalization, and signature segmentation to separate strokes (using pen-down and pen-up information). The verification is done in two levels. Firstly the total writing time is computed by finding the first pen-down and the last pen-up position in the signal. If the total writing time significantly differs from genuine signature it is rejected. Secondly if the tested signature pass the first test time series modeling with *autoregresive* (AR) technique is used to calculate the AR coefficients from all the segments. The AR coefficients are estimated using Levinson-Durbin or Burg method. These coefficients are then used to obtain the *power spectral density* (PSD) to represent each segment. Combination of all the PSD values from all the segments represent the signature.

Other neural network were proposed in [44] for off-line signature verification. An EER 10% on a data set of over 3 000 test samples is reported. The main point of the approach described in [44] is the following: Geometric features of input signature image are simultaneously examined under several scales by a neural network classifier. An overall match rating is generated by combining the outputs at each scale. Artificially generated genuine and forgery samples from enrolment reference signatures are used to train the neural network, which allow definite training control and at the same time significantly reduce the number of enrolment samples required to achieve good performance.

[19] Here it is set up manually which is considered to be the worst case.

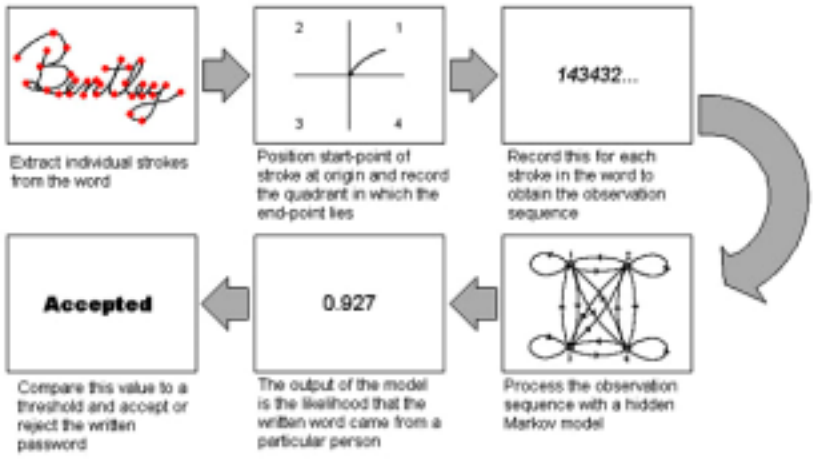

Figure 4.20: The MM based verification

The use of Markov models is reported in [135]. Markov models (MMs) are quite good at separating a signal (in this case a stream of handwriting) into a series of frames or states, and then comparing the properties of these with corresponding states extracted from other handwriting samples. A MM is a stochastic model which describes the probability of moving from one state to another based on the transition probabilities and the previously visited states. The underlying assumption of the MM is that the signal can be well characterized as a parametric random process, and that the parameters of this process can be estimated in a precise, well-defined manner [101].

The MM based system reported in [135] consists of just five different symbols. These are four simple directional symbols (each of these is represented by a single state in the *ergodic*[20] Markov model) and a single symbol representing a "pen-up". Also, in an attempt to illustrate the length of time spent in a particular state, the current observation symbol is repeated every twenty-fifth of a second that the system remains in that state. A particularly long stroke occurring in state one, for example, would result in a series of 1's in the observation sequence. The system therefore is recording the basic shape of the handwriting as well as including some sense of timing and velocity in the observation sequence. The process of extracting the observation symbols and computation of distance measure between test and reference signature (calculated based on multiplying probability values in the Markov model) is illustrated in Figure 4.20.

On-line signature verifier using a combination of Kohenen self-organizing feature map (KSOFM) and hidden Markov models (HMM) have been reported [56]. Experimental result yield FAR of 15% while FRR was set to zero. For data acquisition have been used a high quality digitizer providing a stream of five-dimensional vectors consisting of x and y coordinates (sampled with resolution 100dpi), pressure and polar angle indicating the way the pen is being held.

After the normalization the velocity and acceleration is computed using derivative of cubic B-spline fitted to the positional data. The pen movement

[20]Each state can be reached from all the other states in a finite number of steps.

angle is calculated by the arc tangent of the B-spline parametric derivative mentioned above (that is why normalization in a rotation invariant way is required).

Two different model structures (as discussed in [125] have been tested: *ergodic model* and *left-to-right model*[21].

Two semantically different approaches were tested using ergodic model. For the first, KSOFM is used to discover *cluster centers* in the training patterns for a specific signer. The n most representative centers are then used to configure n HMM states' density functions. A Baum-Welch training phase is then used to further discover the model parameters from the training signature set. The training set and a separate test set is used to establish a log likelihood interval for specific model and the FRR can be dictated by using an appropriate lower bound log likelihood as a threshold value. An attractive feature of the clustered ergodic approach is that a small model (less then ten states) is used to decribe a signature with relative little performance degrading.

The second ergodic model is based on angular intervals of strokes. A number of states n, is chosen prior to training which are used to divide 360° into n angle intervals. To initialize the model, n KSOFMs are the used to discover centers in the signature data-groups formed by inspecting the derived pen movement angle component. Again, a Baum-Welch training phase is used to refine the parameters.

The idea of employing a left-to-right model is that each state should handle a segment of the signature data with smaller state indices conforming to smaller boundary time values in the signature data. The number of segments n, is selected beforehand resulting in n states in the left-to-right model. For the initialization KSOFMs are used to discover the data centers in each frame after which Baum-Welch training phase refines the parameters.

The FRR was set to zero by selecting the smallest log likelihood resulting from matching the training and testing sets against the model. Reported results (depending on the method used) are 13–15% FAR. The test using randomly initialized HMM, resulted in a FAR of 40% which reinforces the importence of prior knowledge injection into generic modeling techniques.

The large contribution to the FAR by a relatively small number of signers, confirms the view shared by various researchers that some signatures are easier to forge than others as different modeling approaches struggled with the same signatures. This fact is a strenuous test of the discriminative power of the handwritten signature modeling technique.

The use of regional correlation method is reported in [127], [88], [24]. The idea behind this algorithm is to cut the signal into regions and to correlate corresponding regions over different time lags to find the best possible match for each pair of regions. According to [127], the handwritten signature is more stable during the pen-down parts of the signature and so, in this context, acceleration signals have been segmented using the pen-down pen-up transitions, rejecting the pen-up parts of the signature. However, this method often results in two incompatible lists of regions for the reference and test signals because pen-lifts are not always detected accurately and also because signers are not always consistent.

Some researchers [88] does not use pen-lift information because they assume

[21] As time increases, the state index increases or stays the same but never decreases.

(based on the hypothesis that the signature is a learned process, that is the result of a ballistic motion with essentially no visual feedback) that the pen-up parts of the signature should be almost as stable as the pen-down parts and certainly harder to imitate. The segmentation process is based only on the observation that handwriting signals tends to fall out of phase beyond a certain time interval. Hence, both the reference and test signals are cut into an equal number of regions. For given signal, all regions are of the same length and regions are correlated in pairs over all allowed time lags. The regional correlation algorithm is invariant to linear transformation on the input signals because of its similarity measure—linear correlation, Furthermore, this invariance is local to each region. This means that neither scale nor offset will affect its result.

In terms of time invariance, regional correlation is somewhat limited to the synchronization of the different regions. This synchronization will accommodate minor hesitation in the execution of the signature, but will not compensate for linear or nonlinear time warping within the regions.

Implementation of DTW by [88] is inspired by work [109] in the field of speech recognition. Authors use a warping function which maps, in sequence, samples points from a reference signal to sample points of a test signal. The distance between two signals is defined by dynamic programming equations given in [88]. These equations respects the usual monotonicity, continuity, and boundary conditions. The distance measure between two sample points used by the dynamic time warping algorithm is very sensitive to scale. However, an offset on the complete signal will not affect the result because the distance measure is centered around the means. With respect to its invariance to timing fluctuations, the algorithm can accept both linear and nonlinear transformations.

Tree matching is a method which estimates the distance between two signals by the distance between their corresponding trees. The tree representation of a waveform is a description of the succession of peaks and valleys in the waveform and of their self-embedded structure. In [17] two new types of trees were introduced—the skeletal tree and the complete tree. In [88], only skeletal trees are considered. The distance between two trees is estimated in terms of node operations which can be linked directly to transformation of the peaks and valleys. Four types of operations on tree nodes are defined for transforming one tree into another. The minim number of those operations is used as a dissimilarity measure. The operations are: father–son splitting and merging, and brother–brother splitting and merging. For skeletal trees, a father–son split or merge corresponds to the growing or shortening of a peak by one quantization interval. A brother–brother split or merge correspond to the deep-ending or shallowing of a valley by one quantization interval. The algorithm to find the minimal number of operations is given in [65].

The tree matching algorithm when used with skeletal tree does not vary at all with the timing fluctuations of signer. Indeed, the skeletal tree representation contains only the sequence of peaks, their self-embedding structure and their amplitude. The duration of the peaks is not taken into account. Regarding the effect of scale and offset on the input signals, skeletal tree matching is not invariant to either, but can be adapt easily and without great penalty to a local or global offset. The way the algorithm measures the distance between two signals is like counting the number of elementary deformations of peaks and

valleys necessary to transform one signal into other, and this makes it a very attractive paradigm.

Handwritten signature verification method based on Fourier transform of the pen nib position (trajectory) is studied in [58] and 15 harmonics with the highest frequencies (for each signature) are used for verification. The study unfortunately does not evaluate the proposed technique well since only one genuine signature writer was used and 19 forgers tried to forge his signature. This limited evaluation leads to a FRR of 0% and a FAR of 2.5%. Also, the proposal to use the 15 harmonics with the highest frequencies requires that information about the harmonics be stored in the reference signature and it may be best to use only the ten or twenty lowest harmonics. Also, the authors make no use of the velocity and acceleration information which should be just as useful as the positional information if not more so. The approach has potential but much further work is needed.

In [2] authors notes that when the forgery looks very similar to the genuine signature most static verification techniques are ineffective. In such situations it is necessary to capture the signature as a gray image rather than a binary image and consider features like the following:

- vertical position that corresponds to the peak frequency of the vertical projection of the binary image
- vertical position that corresponds to the peak frequency of the vertical projection of the high pressure image (an image that only contains pressure regions above some threshold value)
- the ratio of the high pressure regions to the signature area
- threshold value separating the high pressure regions from the rest of the image
- maximum gray level in the signature
- difference between the maximum and minimum gray levels in the signature
- signature area in number of pixels

The paper [23] presents the idea that a signature consists of a sequence of fundamental components delimited by abrupt interruptions which the authors claim occur in positions that are constant in the signature of each individual, although the number of components in one signature may be different than in another signature of the same subject. These components are called fundamental strokes and the technique presented carries out spectral analysis of the strokes.

The authors describe a technique in which two tables are built based on the reference signatures giving the components found in each reference signature and their sequence. Since the number of components in different signatures of the same subject can be different a clustering technique is used to find which components are there in a signature and a sequence of these components is built. The verification involves finding the components of the test signature by

using clustering and then checking that the components appear in the sequence derived from the reference signatures. If the sequence does not fit, the test signature is rejected otherwise the components are compared with those of the reference signatures.

An overall FRR of 1.7% and FAR of 1.2% was obtained. The paper notes that Fourier analysis of components was also used but details are not provided. The major problem with the evaluation is the use of 50 reference signatures which is quite unrealistic.

Multilevel signature verification system that uses global features as well as point-to-point comparison using personalized thresholds is presented in [93]. Global features (includes the percentage of penup time and the percentage of time when the angular velocity is positive) are used for the first stage of the verification. The signature is normalized using rotation and scaling and local correlations are computed between portions of the test signature velocity values with the corresponding values of the reference signature using segments alignment using elastic matching. This second stage is followed by a third stage involving calculation of variations between the normalized coordinate values of the test signature and the reference signature using local elastic pattern matching.

Concluding Remarks

Details about the values of the various features for the genuine signatures and the forged signatures are presented in [42]. It is shown that feature values for most forged signatures are quite random and therefore some by chance will happen to be close to the reference mean. Fortunately though if a forged signature has a feature value close to the reference mean for one feature, it is often not close to the mean for another feature. There are two important points arising from [41] that should be stressed:

- A surprising number of forged signatures have feature values that are more than twenty standard deviations away from the reference signature mean. Many are even more than 50 standard deviations away from the reference signature mean. This would not have been surprising for random forgeries but all these forgeries are skilled forgeries produced by volunteers who had practiced forging the signatures.

- Feature values of many forged signatures were far away from the reference signature means for the following features: total time, the acceleration sign changes, the penup time, path length, and the x–acceleration zero count. For other attributes (the two velocity sign changes, y–acceleration zero count and the segment count), many more forged signatures had feature values closer to the corresponding reference signature mean.

An interesting ideas can be derived from [21] where authors allow up to *three trials* for signature verification and a false rejection occurs only if all the three signatures fail the verification test. This, of course, disables comparison of their system with others since the definitions of FRR and FAR are not what are normally used in the literature.

Authors also discuss the possibility of using personalized feature sets for each person rather than using the same features for all persons. Some evidence is presented that personalized feature sets can improve the performance of a signature verifier.

Finally, authors awarded prizes for best forgeries which motivated volunteers to produce better forgeries.

In a US patent [89] it is proposed that a reference signature consisting of means and standard deviations of the features used and at least six sample signatures be collected. It is noted that if six sample signatures are gathered under identical conditions, the standard deviations might be too small to be an accurate estimate of the standard deviations of the persons signatures. It is therefore proposed that the standard deviations be modified to be the means of the standard deviations of the individual and the standard deviations of the population at large. The resulting values should be used if they are found to "conform to what experience shows to be realistic limits". If the values are not "realistic" further sample signatures may be obtained and some of the previously obtained sample signatures may be discarded if they were widely inconsistent with the other sample signatures. In some cases, the whole attempt to obtain a reference signature may be aborted and a new set of sample signatures obtained at another occasion.

A number of other suggestions have been made including different values of thresholds for different individuals and the possibility of basing the threshold on the value of the merchandise being bought and the credit rating of the person. Suggestions are also made about updating of the reference signature by applying a weighting of 90% to stored parameters and 10% the new ones obtained from the test signature. The patent also includes a list of 99 features that may be used for automatic handwritten signature verification. It is suggested that an optimum number of features is between 10 and 20 but higher number may be desirable in some situations and somewhat different parameters may be used in different instances of the same signature verification system.

As was touched on earlier, the state of the art in handwritten signature verification (HSV) makes it impossible to draw definitive conclusions about which techniques are the best since:

- Performance of a HSV system that uses different features for different individuals is better than a system that uses the same features for all.
- Performance of a HSV system that uses different threshold values for different individuals is better than a system that uses the same thresh old value for all.
- Performance of a HSV system that uses more signatures for building the reference signature is better than a system that uses a smaller number of signatures.
- FRR of a HSV system that uses more than one reference signature to make a judgement about whether the subject is genuine or not is better than a system that uses only one signature.
- Performance of a HSV system that uses the genuine signatures as well as some or all the forgeries that are used in performance evaluation in

building the reference signature or in deciding which features to choose and/or what threshold to select is better than a system that does not use any test signatures in building the reference signature.

- Performance of a HSV system that is tested on only a small database of test signatures that has signatures from only a small number of subjects is likely to be better than a system that uses a larger database which has signatures from a larger number of subjects.
- Performance of a HSV system that is tested on a database of test signatures that were screened to eliminate some subjects that had problem signatures is likely to be better than a system that has not carried out any such screening.

The survey seems to indicate that any technique using statistical features is unlikely to provide a total error rate (FAR + FRR) of less than 10% if a reasonably large signature database is used. Most research work that claims much better results have been found to have weaknesses in their performance evaluation.

The best techniques are likely to be based on using a combination of statistical features as well as the shape of the signatures.

(5) Current and Future Work

The standard methods mentioned in Section 4.4 are more or less founded on thresholding. However, the problem of the automatic setup of the threshold has not yet been solved. Intensive research into training algorithms capable of setting up the threshold less dependent on or possibly totally independent of the human factor is needed. Such an ideal analytical method would accept several training samples and produce an appropriate threshold (or threshold vector).

The following section summarizes briefly the author's work (in progress) concerning the design and development of handwriting analysis methods applied to the BiSP[1] pen and gives an outline of the content of his doctoral thesis.

5.1 Innovative Data Acquisition Devices

Commercial systems designed for handwritten text acquisition use as an input device a scanner, a pen with a tablet or a GPS-based pen with a (infrared or ultrasound) transmitter and several receivers (Section 4.2). The obvious disadvantage of these devices is the limited mobility of a system composed of two or more parts. Systems based on optical (OTM) technology require additional light sources (usually built-in inside the pen) in order to work properly which is intrusive and uncomfortable.

Under the BiSP project several pen prototypes were constructed. These prototypes integrate all the electronic devices needed for the data acquisition inside the pens and are ergonomic and non-invasive as they do not emit light, sound, or electromagnetic radiation and provide a comfortable feeling while writing.

BiSP pens can compete with all the pen products mentioned in section 4.2 in the area of transferring handwritten information from paper to a computer. The products differ in the technologies they use and the features the products provide to the users.

The BiSP pen has two significant advantages: It makes it is possible to write with the pen even without having access to a computer (e.g. on a train or during a lecture) and without need of any special paper or installing any other hardware (tablet, IR receivers, etc.)—from this point of view the BiSP pen is unique. Of course, while writing with the pen both the electronic and paper versions of the document are available at once.

5.1.1 Acceleration Sensor Pen

The first pen prototype (Fig. 5.1) consists of two sensors integrated in a pen producing a total of three signals (Fig. 5.2). The acceleration sensor—accelerometer

[1] BiSP—Biometrical Smart Pen for Personal Identification is a joint project of University of West Bohemia in Pilsen and University of Applied Sciences in Regensburg

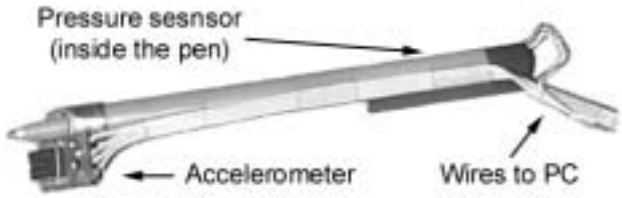

Figure 5.1: Acceleration sensor pen

placed near the pen nib produces two signals corresponding to the horizontal and vertical movements of the pen. A pressure sensor—based on the piezoelectric effect—is built into the pen (see the two thin wires on the right-hand side in Fig. 5.1).

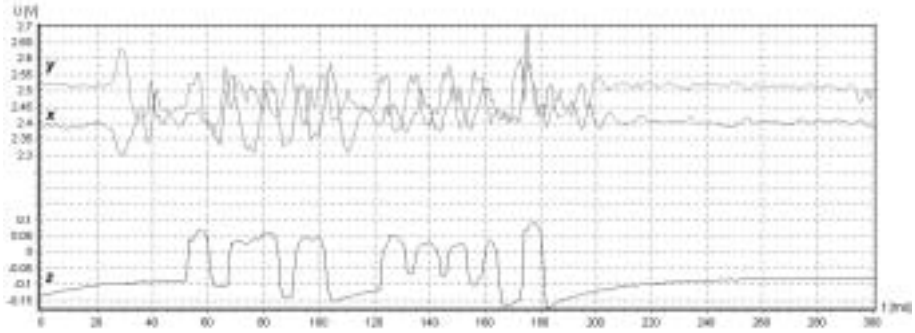

Figure 5.2: Signals of acceleration and pressure

The accelerometer used in the pen is an ordinary commercial integrated circuit made by Analog Devices. ADXL202 is 2-axis acceleration sensor that uses a mass that moves between the two pairs of capacitors. The sensor output signals are derived from the change in capacities. The two pairs of capacitors are orthogonal to each other, so the obtained signals correspond to the acceleration in x and y axes (pair of signals with high values in Fig. 5.2). In fact, this is true if the pen is held in the proper position (Fig. 5.3). If the pen is slightly rotated, then the x axis does not correspond exactly to the left-to-right direction of writing (direction of base-line) and the y axis does not correspond to the direction orthogonal to the base-line. This problem can be solved by working with polar coordinates (amplitude and phase) computed from the x and y signals. The accelerometer is able to measure both the dynamic acceleration (movement of pen, vibration) and static acceleration (Earth movement acceleration). As the measurement of the static acceleration cannot be avoided, the elimination of the influence of gravity on the data is necessary. The accelerometer has been designed to be used in industrial applications—not in handwritten text analysis applications. Therefore, the signals produced by the sensors are not suitable for character recognition although they are feasible for on-line signature verification because for signature verification the signature dynamics is more important than its appearance. The author carried out a number of experiments trying to

reconstruct the trajectory of the pen by computing the velocity and the position out of the acceleration signals, but the results were not good. This transformation works only if the signatures (or gestures) are very large (as large as a sheet of paper) and even if they are so large, results are not very good.

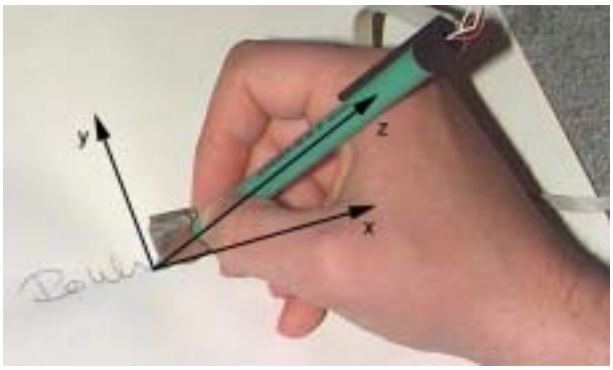

Figure 5.3: Writing with the pen

The pressure sensor used in pen prototype (the PSt150/2x3/7 by Piezomechanik GmbH) produces much better signals. It can be easily identified when the pen was in contact with the paper and when not. For example the pressure signal (the one with the lowest values in Fig. 5.2) corresponds to the signature with five separate parts, where the fourth part is the longest and consists of four characters. After the end of each of the five parts there is a slight decrease in voltage, caused by the capacitor added to the pressure signal in order to stabilize it. Diacritical marks as well as parts of the signature where the trajectory changes radically can be recognized.

5.1.2 Pressure Sensor Pen

Second prototype of the BiSP pen consists of pair of pressure sensors that measure the horizontal and vertical movements of the pen nib and pressure sensor that is placed in the top part of the pen. The pen produces a total of three signals (Fig. 5.5). The upper signal corresponds to the pressure sensor (the same as is used in the first prototype of the BiSP pen) and the other two correspond to the horizontal and vertical movements of the pen. The data (Fig. 5.5) were acquired while writing the word "Dobrou" (Fig. 5.6), which means "good".

Four sensors that measure the horizontal and vertical movements of the pen are located near the pen nib and are placed orthogonal to each other. The signal produced by the horizontal pair of sensors is called x and the one produced by the vertical sensors y. Each pair of sensors is connected to a Wheatstone bridge. Therefore there is only one output signal corresponding to the horizontal movement of the pen (x) and one corresponding to the vertical movement (y). Unfortunately it is not possible to give more details about the design and implementation of the pen as it is currently in patent pending status.

In the following paragraphs the signals (Fig. 5.5) will be discussed in greater detail. Note the behavior of the pressure signal—the test person wrote the word (Fig. 5.6) in two parts—first the letter "D" and then the rest of the word

Figure 5.4: Pressure sensor pen—with and without cover

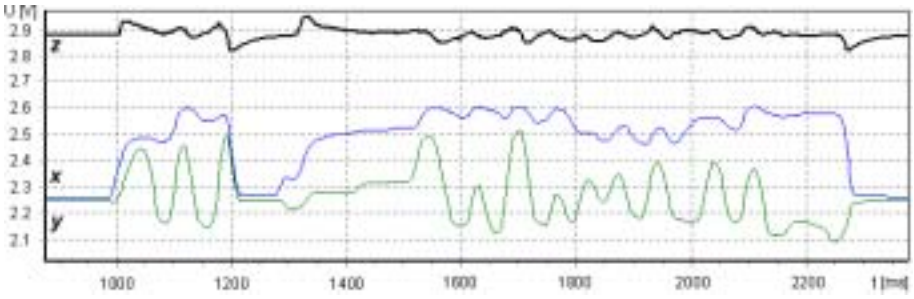

Figure 5.5: Signals produced by the pressure sensor pen

"obrou". The beginning of the writing as well as the end can easily be identified using the first-order difference of the pressure signal.

Note also that the x signal (corresponding to the horizontal movement) does not fall below the quiescent value. There are two reasons for this. Firstly, the test person held the pen slightly slanted to the right, therefore one of the sensors measuring the nib movement was still under certain pressure. Secondly, the word was written from left to right, hence the left sensor was stimulated much more than the right one.

Figure 5.6: Word "Dobrou"

5.2 Handwriting Analysis Methods

5.2.1 Clustering Approach to Signature Verification

The main problem of handwritten signature verification that researchers encounters is the fact that signature verification differs from the general classification problem—the goal of the general classification problem is to choose one class from several classes, whereas the training data contain specimens from all

classes. For signature verification all the training data are just genuine signatures. No training data for the second class (forgeries) are available.

The basic idea of verification algorithm proposed is straightforward—compute the distance between the tested signature and the pattern. If the distance is small then the tested signature is probably genuine. Now the problem is reduced to decision "What distance is small?". That distance is called *critical cluster coefficient* and is computed as a mean mutual difference between all pairs of patterns in the class. It means that the critical cluster value describes the similarity of signatures. For authors whose signatures are nearly the same this coefficient is low; in other words a signature is classified as genuine if it is very similar to some pattern. In contrast, if the patterns are not uniform then the chance that the tested signature will be recognized as genuine is much greater because of the higher value of the critical cluster coefficient.

```
For each class C
    For each feature f
        For each pair of signatures Classes[C][i] and Classes[C][j]
             Compute the difference between Classes[C][i] and
             Classes[C][j] and add it to an extra variable Sum[f]
        Compute mean value mean[f] and variance var[f] of each
        feature over all pairs using the variable Sum[f]
    Compute critical cluster coefficient using variances var[f]
    and weights w[f] over all features f
```

Algorithm 5.1: Signature verification—training phase

```
For class C to be verified
    For each pattern Classes[c][i]
        For each feature f
            Compute the difference and remember the least one
            over all patterns
    Sum up products of least differences and weights w[f]
    and compare the sum with Critical cluster coefficient
```

Algorithm 5.2: Signature verification—testing phase

As usual, recognition methods are not applied directly to raw data but the preprocessing and feature extraction methods are used to reduce the number of values representing an object. The features used are based on statistical characteristics of signatures such as "maximum value of pressure signal" or "variance of acceleration signal amplitude". The recognizer uses a total of 20 features so far—each of them having a different weight in classification since some features are better than others. The features of each class create clusters in a n-dimensional space and critical cluster coefficient describes the "compactness" of the cluster.

Depending on the size of the training and testing data, the accuracy achieved yield 1.3% EER on skilled forgeries [106].

5.2.2 Application of ART-2 Neural Network to Signature Verification

As already mentioned the frequently used supervised learned neural network model such as *multi-layer perceptron* (MLP) can hardly be applied to the signature verification since the training data are from on class only (genuine signatures). The *adaptive resonance theory* (ART), developed by Carpenter and Grossberg, have been designed for clustering of binary input vectors (ART-1) or continuous-valued input vectors (ART-2). With regards to the features that are used for description of signals, the ART-2 model is applicable to signature verification. The general architecture and description of the ART-2 network is presented in [35], [13].

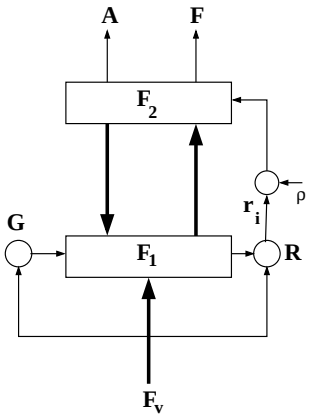

Figure 5.7: ART-2 neural network

The basic structure of the network verifier is illustrated in Fig. 5.7. The network consists of two layers of processing elements labelled F_1 (input and interface units) and F_2 (cluster units), both fully interconnected with the other, and supplemental units G and R (called *gain control unit* and *reset unit*), which are used to control the processing of the input vector and creating of the clusters.

The input and interface layer F_1 consists of six sub-layers (these are not illustrated in Fig. 5.7) and each sub-layer has the same number of processing units. The purpose of these sub-layers is to allow the ART-2 network to process continuously varying inputs. Moreover, they normalize the components of the feature vector and suppress the noise. The size of the F_1 layer (and hidden sub-layers) is set to 19 and corresponds to the size of the feature vector.

The clustering layer F_2 consists of two processing units only, the former (labelled A) is active only if the feature vector corresponding to the genuine signature appears at the input of the network, the latter (labelled F) is active in other cases. More clusters are not applicable in signature verification application.

The slow learning mode is used for ART-2 network training. The parameters of the hidden sub-layers of F_1 and vigilance parameter ρ were set so that only the unit labelled A of layer F_2 was active during the whole training procedure (the network places the template signatures only in one cluster and adapts the corresponding weights between F_1 and F_2 layers).

When the training is completed, the network is prepared for verification. The parameters of F_1 sub-layers are not changed during the verification, only the vigilance parameter ρ have to be set properly in order to set the genuine signatures and forgeries to right clusters. The vigilance parameter ρ can be set manually or automatically according to number of training vectors and activation levels of unit R.

For feature extraction the *fast wavelet transform* (FWT) is used [97]. For decomposition the Daubechies and Coiflet wavelet families were tested, the 5^{th} order Daubechies wavelet yielded the best results [75]. Writers were instructed to evaluate their signatures by a rating on scale 1 – 4 (1 for best signature; 4 for abortive). For verifier training, only the five signatures labelled be rating 1 or 2 were chosen. The EER yields 9% on skilled forgeries in case of automatic setting of vigilance parameter and is slightly better if set manually.

5.2.3 Other Handwriting Analysis Methods

In order to prove that the BiSP pens (Section 5.1) have the potential to be commercially successful even in different applications than signature verification several other experiments were carried out: author identification and handwritten text recognition.

Author identification is a problem slightly different from the signature verification although it looks similar at the first sight. The diversities are the following: samples are classified into several classes (each class corresponds to one author) and the written word is not a signature but any other word—usually the same word for all authors. Within the experiment the knowledge of graphologists and forensic experts was exploited (Section 3). Unfortunately, not all of these peculiarities and characteristics can be used for automatic writer identification—many of them require larger text samples than a single word. Sufficient for accurate classification proved to be characteristics corresponding to expansion in height, coordination, speed, tension and rhythm.

Neural network (*multi-layer perceptron*) has been used as a classifier and has been trained using a variant of the back-propagation algorithm with momentum. Results obtained from the testing data of 20 words per author yield cca 95% accuracy depending on the size of training set (even 100% accuracy was achieved by a well-trained network) [107].

The recognition of handwritten characters and handwritten words using the BiSP pen data is a challenging task as the data obtained are entirely different from the data obtained by other input devices currently used (Section 4.2). All the state-of-the-art systems (both off-line and on-line) utilize the pen nib coordinates (either as two-dimensional image or as a sequence of samples) for recognition. Unfortunately, the BiSP pen data cannot be transformed into x and y coordinates because of hardware construction restrictions.

The dynamic time warping (DTW) method has been tested for the recognition of handwritten characters and handwritten words, but the results were not good. Moreover, DTW can hardly be used for a large vocabulary which makes this method unsuitable for possible commercial utilization.

That is the reason why the author decided to use hidden Markov models (HMM) for handwritten word recognition that are reported to provide good

results (Section 4.3.1). The first experiments on the corpora containing 2 000 words yield a 90% word recognition rate. Taking into account the restrictions mentioned above and the fact that the recognition methods already published cannot be simply adopted, it is a promising result [108]. As the quality of the BiSP pen signals increases throughout the development of the pen's hardware even better results can be expected.

5.2.4 Advanced Means of Signature Verification

The standard methods (as mentioned in Section 4.4) are more or less based on the *global weights* which are believed to represent the differentiation ability of features in feature vectors. This means that these weights (that are set up during the system development and are based on the experimental tests) are shared by all users (signers). In contrast to this the latest experiment indicates that the differentiation ability varies user by user. Therefore, the call for a *local weights* methods means that more user-adaptive methods are needed. A possible solution of this problem could be the modification of the critical cluster coefficient computation (Section 5.2.1). This approach is now being researched and the first results are very encouraging.

A robust system that is capable to provide both accommodating and reliable framework for person identification based on signature verification is being developed. This system should include following features:

Adjustable Security Level: The system should allow users to adjust the security level for better protection against would-be forgers. This is possible through adjustment of the FAR and FRR ratio parameter. The user may wish, for example, to set up the FAR to 0 at the cost of higher FRR.

Progressive Thresholding Strictness: The user of the system should have the ability to enable *progressive thresholding strictness* which is a method that steps up security measures depending on, for example, the amount of money to be withdrawn from the bank or the value of information to be allowed to access.

Optional Self-Learning Mode: It has been proved that with the passage of time, a person's signature will change together with physiological changes of the person. The system should provide an optional self-learning mode to cater to these changes.

Training Set Size: Researchers often complain that the size of the training set is insufficient—in banks, for example, during the enrolment process it would be annoying to ask the client to sign five times just to collect enough data for highly reliable verification. However, the client signs not only the *signature card*, but also signs the agreement, certificate of initial deposit, etc. These would give enough data for the verification purpose.

Database Theft Countermeasure (Listening Device Detection): As it is virtually impossible for a person to sign his or her name twice in exactly the same way, an absolutely identical signature has to be a copy, which can be easily detected. Identical signatures will be treated as a security threat and rejected.

Signature Evaluation: The writer knows his handwriting very well and therefore he or she can rate the signatures (during the enrollment process) to express which signatures are suitable for building the training set and which are not. This method could improve the FAR considerably.

Open System Architecture: The system should be ready to incorporate other biometrics that are able to support the accept/reject decision. These non-intrusive biometrics include hand geometry, speaker identification, face recognition, and eventually also off-line signature verification.

5.3 Aims of Doctoral Thesis

The aims of author's doctoral thesis based on the previous sections can be summarized as follows:

1. Thorough research into the design, shape, and structure of handwriting along with a comparative study and investigation of other handwriting habits and patterns, which may lead to the design of new methods for signature verification.

2. The application of and their evaluation in various conditions within the framework of the BiSP project.

3. The implementation of newly proposed methods and the evaluation of their efficiency and contribution to the development of the signature verification domain.

4. Further improvements in the design and development of the identity verification system outlined above in section 5.2.4 as well as an implementation of the modules of this system.

Bibliography

[1] S. Abramowsi, H. Müller: **Geometrisches Modeliren**. Reihe Informatik, Vol. 75, BI Wissenschaftsverlag, Mannheim, Wien, Zürich, 1991.

[2] M. Ammar, Y. Yoshida, T. Fukumara: **A New Effective Approach for Offline Verification of Signatures by Using Pressure Features**. IEEE Transactions on Systems, Man and Cybernetics, Vol. 16, No. 3, 1986.

[3] H. S. M. Beigi, S. H. Maes, U. V. Chaudhari, J. S. Sorensen: **IBM Model-Based and Frame-by-Frame Speaker Recognition**. In Speaker Recognition and its Commercial and Forensic Appications, Avignon, 1998.

[4] D. Besner, G.W. Humphreys: **Basic Processes in Reading: Visual Word Recognition**. Hillsdale, New Jersey, 1991.

[5] W. Bicz, Z. Gurnienny, M. Pluta: **Ultrsound Sensor for Fingerprints Recognition**. In Proc. of SPIE, Vol. 2634, Optoelectronic and Electronic Sensors, 1995.

[6] P.E. Bramall, C.A. Higgins: **A Cursive Script-Recognition System Based on Human Reading Models**. Machine Vision and Applications, Vol. 8, No. 4, 1995.

[7] J. Brault, R. Plamondon: **A Complexity Measure of Handwritten Curves: Modeling of Dynamic Signature Forgery**. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 2, 1993.

[8] L. Breiman et al: **Classification and Regression Trees**, The Wadsworth Statistics/Probability Series, Wadsworth, Belmont, 1984.

[9] T. M. Breuel: **A System for the Off-line Recognition of Handwritten Text**, In Proc. of the 12^{th} International Conference on Pattern Recognition, Jerusalem, 1994.

[10] R. Brunelli, T. Poggio: **Face Recognition: Features Versus Templates**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, 1993.

[11] H. Bunke, M. Roth, E. G. Schukat-Talamazzini: **Off-line Cursive Handwriting Recognition Using Hidden Markov Models**. Pattern Recognition, Vol. 28, No. 9, 1995.

[12] T. Caesar, J. M. Gloger, E. Mandler: **Preprocessing and Feature Extraction for a Handwriting Recognition System**. In Proc. of the 2^{nd} IAPR Int Conf. on Document Analysis and Recognition, Tsukuba Science City, 1993.

[13] G.A. Carpenter, S. Grossberg: **ART-2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns**. Applied Optics, No. 26, 1987.

[14] W. F. Clocksin, M. Khorsheed: **Word Recognition in Arabic Handwriting**. In Proc. of the 8^{th} Intrenational Conference on Artificial Intelligence Applications, Volume 1, Cairo, 2000.

[15] R. Chellappa, C. L. Wilson, S. Sirohey: **Human and Machine Recognition of Faces: A Survey**. Proceedings of the IEEE, Vol. 83, No. 5, 1995.

[16] m. Chen, A. Kundu, J Zhou: **Off-line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 5, 1995.

[17] Y. C. Cheng, S. Y. Lu: **Waveform Correlation by Tree Matching**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, No. 3, 1985.

[18] W. Cho, S. Lee, J. Kim: **Modeling and Recognitin of Cursive Words with Hidden Markov Models**. Pattern Recognition, Vol. 28, No. 12, 1995.

[19] C.F. Colmas: **The Writing Systems of the World**. Blackwell, 1980.

[20] M. Côté, E. Lecolinet, M. Cheriet, and C.Y. Suen: **Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concept**. International Journal on Document Analysis and Recognition, Vol. 1, 1998.

[21] H. D. Crane, J. S. Ostrem: **Automatic Signature Verification using a Threeaxis ForceSensitive Pen**. IEEE Transactions on Systems, Man and Cybernetics, Vol. 13, No. 3, 1983.

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

[23] G. Dimauro, S. Impedovo, G. Pirlo: **ComponentOriented Algorithms for Signature Verification**. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, 1994.

[24] G. Pirlo: **Algorithms for Signature Verification**. In Fundamentals in Handwriting Recognition, Bari, 1993.

[25] G. Dimauro, S. Impedovo, G. Pirlo: **On-line Signature Verification by a Dynamic Segmentation Technique**. In Proc. of IWFHR-3, Buffalo, 1993.

[26] J. G. A. Dolfing: **Handwriting Recognition and Verification: A Hidden Markov Models Apporach**. Ph.D. thesis, Eindhoven University, 1998.

[27] J. G. A. Dolfing, E. H. L. Aarts, v. J. J. G. M.: **On-line Signature Verification with Hidden Markov Models**. In Proceedings of the International Conference on Pattern Recognition, 1998.

[28] B. Duc, E. S. Bigün, J. Bigün, G. Maître, S. Fischer: **Fusion of Audio and Video Information for Multi Modal Person Authentication**. Pattern Recognition Letters, Vol. 18, No. 9, 1997.

[29] H. Dullink, B. van Daalen, J. Nijhuis, L. Spaanenburg, H. Zuidhof: **Implementing a DSP Kernel for Online Dynamic Handwritten Signature Verification Using the TMS320 DSP Family**. EFRIE, France 1995.

[30] G. J. Edwards, C. J. Taylor, T. F. Cootes: **Interpreting Faces using Active Appearance Models**. In International Conference on Face and Gesture Recognition, No. 3, 1998.

[31] A. J. Elms: **The Representation ond Recognition of Text Using Hidden Markov Models**. Ph.D. thesis, University of Surrey, 1996.

[32] B. G. et al.: **Issues in Large Scale Automatic Biometric Identification**. In IEEE Workshop on Automatic Identification Advanced Technologies, Stony Brook, 1996.

[33] C. Falure, E. Lecolinet: **OCR: Handwriting**. In Survey of the State of the Art in Human Language Technology, 1995.

[34] C. Farouz, M. Gilloux, J.-M. Bertille: **Handwritten Word Recognition with Contextual Hidden Markov Models**. In Proc. of the IWFHR-6, Taejon, 1998.

[35] L. Fausett: **Fundamentals of Neural Networks**. Prentice-Hall, New Jersey, 1994.

[36] D. T. Follette, E. B. Hultmark, J. G. Jordan: **Direct Optical Input System for Fingerprint Verification**. IBM Technical Disclosure Bulletin: 04-74, 1974.

[37] S. Furui: **Recent advances in Speaker Recognition**. In Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science 1206, Springer-Verlag, Berlin, Heidelberg, 1997.

[38] A. Goddard: **Disappointing Verdict on Signature Software**. New Scientist, Vol. 142, 1994.

[39] D. Guillevic, C. Y. Suen: **HMM Word Recognition Engine**. In Proc. of the 4^{th} ICDAR, Ulm, 1997.

[40] G. K. Gupta, R. C. Joyce: **A Simple Approach to Dynamic Hand-Written Signature Verification**. Technical report, James Cook University of North Queensland, 1995.

[41] G. K. Gupta, R. C. Joyce: **A Study of Some Pen Motion Features in Dynamic Handwritten Signature Verification**. Technical report, James Cook University of North Queensland, 1977.

[42] G. K. Gupta, A. McCabe: **A Review of Dynamic Handwritten Signature Verification**. Technical report, James Cook University, 1997.

[43] K. Hayashi, E. Okamoto, M. Mambo: **Proposal of User Identification Scheme Using Mouse**. Lecture Notes in Computer Science 1334, Springer-Verlag, Berlin, Heidelberg, 1997.

[44] K. Huang, H. Yan: **Off-Line Signature Verification by a Neural Network Classifier**, Proc. of the 17^{th} Australian Conference on Neural Networks, Canberra, 1966.

[45] L. P. Heck, M. Weintraub: **Handset-Dependent Background Models for Robust Text-Independent Speaker Recognition**. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997.

[46] N. M. Herbst, C. N. Liu: **Automatic Signature Verification Based on Accelerometry**. Technical report, IBM Watson Research Center, 1977.

[47] H. Hild, A. Waibel: **Speaker-Independent Connected Letter Recognition with a Multi-State Time Delay Neural Network**. In proceedings of 3^{rd} European Conference on Speech, Comunication and Technology, Volume 2, Berlin, 1993.

[48] O. Hilton: **Signatures—Review and a New View**. Journal of Forensic Sciences, Vol. 37, No. 1, 1992.

[49] S. Jaeger: **Recovering Dynamic Information from Static, Handwritten Word Images**. Ph.D. thesis, Ulm, 1998.

[50] S. Jaeger, S. Manke, J. Reichert, A. Waibel: **On-line Handwriting Recognition: The NPen++ Recognizer**. International Journal on Document Analysis and Recognition, Volume 3, 2000.

[51] A. Jain, R. Bolle, S. Pankanti: **BiometricsPersonal Identification in Networked Society**. Kluwer Academic Publishers, Boston, 1999.

[52] S. Jung, R. Thewes, T. Scheiter, K. F. Gooser, W. Weber: **A Low-Power and High-Performance CMOS Fingerprint Sensing and Encoding Architecture**. IEEE Journal of Solid-state Cicuits, Vol. 34, No. 7, 1999.

[53] A. Kaltenmeier, T. Caesar, J. M. Gloger, E. Mandler: **Sophisticated Topology of Hidden Markov Models for Cursive Script Recognition**. In Proc. of the 2^{nd} International Conference on Document Analysis and Recognition, Tsukuba Science City, 1993.

[54] G. Kaufmann, H. Bunke, M. Hadorn: **Lexicon Reduction in a HMM-Framework Based on Quantized Feature Vectors**. In Proc. of the 4^{th} International Conference on Document Analysis and Recognition, Ulm, 1997

[55] J. Kittler, Y. Li, J. Matas, M. Ramos Sánchez: **Lip-shape Dependent Face Verification**. In Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science 1206, Springer-Verlag, Berlin, Heidelberg, 1997.

[56] T. Kohenen: **Self-Organising Maps**. Springer-Verlag, Berlin, Heidelberg, 1995.

[57] S. Y. Kung: **Digital Neural Networks**. PTR Prentice-Hall, Inc., 1993.

[58] C. F. Lam, D. Kamins: **Signature Verification Through Spectral Analysis**. Pattern Recognition. Vol. 22, No. 1, 1989.

[59] F. Leclerc, R. Plamondon: **Automatic signature verification**. International. Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, 1994.

[60] K. F. Lee: **Automatic Speech Recognition, The Development of the SPHINX System**. Kluwer Academic Publishers, Boston, London, 1989.

[61] L. L. Lee: **Online Systems for Human Signature Verification**. Ph.D. thesis, Cornell University, 1992.

[62] R. P. Lippmann: **An Intorduction to Computing with Neural Nets**. IEEE ASSP Magazine April 1994.

[63] C. N. Liu, N. M. Herbst, N. J. Anthony: **Automatic Signature Verification: System Description and Field Test Results**. IEEE Transactions on Systems, Man and Cybernetics, Vol. 9, No. 1, 1979.

[64] G. Lorette: **Online Handwritten Signature Recognition based on Data Analysis and Clustering**. Proc. of the 7^{th} International Conference on Pattern Recognition, Montreal, 1984.

[65] S. Y. Lu: **A Tree Matching Algorithm Based on Node Splitting and Merging**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 2, 1984.

[66] J. Luettin, N. A. Thacker, S.W. Beet: **Speaker Identification by Lipreading**. In Proc. of the 4^{th} International Conference on Spoken Language Processing , Vol. 1, 1996.

[67] S. Maddouri, H. Amiri, A. Belaid: **Local Normalization Towards Global Recognition of Arabic Handwritten Script**. In Proc. of the Document Analysis and Systems Intrenational Conference, Rio de Janeiro, 2000.

[68] J.-F. Mainguet, M. Pegulu, J. B. Harris: **Fingerchip: Thermal Imaging and Finger Sweeping in a Silicon Fingerprint Sensor**. In Proc. of AutoID'99, 1999.

[69] D. Maio, D. Maltoni: **Direct Gray-Scale Minutiae Detection in Fingerprints**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 1, 1997.

[70] E. Mandler, R. Oed, W. Doster: **Experiments in On-line Script Recognition**. In Proc. of the 4^{th} Scandinavian Conference on Image Analysis, 1985.

[71] S. Manke, M. Finke, A. Waibel: **NPen++: A Writer Independent, Large Vocabulary On-line Cursive Handwriting Recognition System**. In Proc. of the International Conference on Document Analysis and Recognition, Montreal, 1995.

[72] S. Manke, M. Finke, A. Waibel: **A Fast Search Technique for Large Vocabulary On-line Handwriting Recognition**. In International Workshop on Frontiers in Handwriting Recognition, Colchester, 1996.

[73] A. J. Mauceri: **Feasibility Studies of Personal Identification by Signature Verification**. Technical report, North American Aviation Co., Anaheim, 1965.

[74] P. Mautner: **Detection of Singular Regions in Fingerprint**. Ph.D. thesis, Univeristy of West Bohemia, Plzeň, 1998.

[75] P. Mautner, O. Rohlík, V. Matoušek, J. Kempf: **Signature Verification Using ART-2 Neural Network**. In proceedings of ICONIP'02 Conference, Singapore 2002.

[76] P. Mautner, O. Rohlík, V. Matoušek, J. Kempf: **Fast Signature Verification without Special Tablet**. In proceedings of IWSSP'02 Conference, Manchester 2002.

[77] B. Miller: **Vital Signs of Identity**. IEEE Spectrum, Vol. 31, No. 2, 1994.

[78] S. F. Mjølsnes, G. Soberg: **A Comparative Performance Experiment of Dynamic Signature Verification Devices**. In Computer Recognition and Human Production of Handwriting, World Scientific, Singapore, 1989.

[79] M. E. Munich, P. Perona: **Camera-based ID Verification by Signature Tracking**. Lecture Notes in Computer Science 1407-1408, Springer-Verlag, Berlin, Heidelberg, 1998.

[80] M. E. Munich, P. Perona: **Visual Signature Verification using Affine Arc-length**. IEEE Computer Vision and Pattern Recognition Volume 2, 1999.

[81] R. N. Nagel, A. Rosenfeld: **Computer Detection of Freehand Forgeries**. IEEE Transactions on Computers. Vol. 26, No. 9, 1977.

[82] V. S. Nalwa: **Automatic On-line Signature Verification**. Proceedings of the IEEE, Vol. 85, No. 2, 1997.

[83] W. Nelson, E. Kishon: **Use of Dynamic Features for Signature Verification**. Proc. of the IEEE International Conference on Systems, Man, and Cybernetics, Charlottesville, 1991.

[84] W. Nelson, W. Turin, T. Hastie: **Statistical Methods for Online Signature Verification**. In International Journal of Pattern Recognition and Artificial Intelligence, Vol.8, No. 3, Singapore, 1994.

[85] A.G. Nijhuis, M.H. Brugge, K.A. Helmholt, J.P.W Pluim, L. Spaanenburg, R.S. Venema, M.A. Westenburg: **Car License Plate Recognition with Neural Networks and Fuzzy Logic**. In Proc. of the ICNN'95, Perth, 1995.

[86] A. S. Osborn: **Questioned Documents**. Boyd Printing Co., Albany, 1929.

[87] W. Peterson, T. Birdsall, W. Fox: **The Theory of Sigmal Detectability**. Transactions of the IRE, Vol. PGIT-4, 1954.

[88] M. Parizeau, R. Plamondon: **A Comparative Analysis of Regional Correaltion, Dynamic Time Warping, and Skeletal tree matching for Signature Verification**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 7, 1990.

[89] J. R. Parks, D. R. Carr, P. F. Fox: **Apparatus for Signature Verification**. US Patent No. 4495644, 1985.

[90] T. Pavlidis: **Algorithms for Graphics and Image Processing**. Springer-Verlag, Berlin, Heidelberg, 1982.

[91] R. Plamondon: **A Renaissance of Handwriting**. Machine Vision and Applications, Vol. 8, 1995.

[92] G. Plamondon: **Progress in Automatic Signature Verification**. World Scientific, Singapore, 1993.

[93] R. Plamondon: **The Design of an Online Signature Verification System: From Theory to Practice**. In International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, Singapore, 1994.

[94] R. Plamondon, G. Lorette: **Automatic Signature Verification and Writer Identification—The State of the Art**. Pattern Recognition, Vol. 22, No. 2, 1989.

[95] R. Plamondon, F.J. Maarse: **An Evaluation of Motor Models of Handwriting**. IEEE Transactions Systems, Man, and Cybernetics, Vol. 19, No. 5, 1989.

[96] R. Plamondon, S. N. Srihari: **On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, 2000.

[97] S. Pitnerr, S.V. Kamarthi: **Feature Extraction from Wavelet coefficients for Pattern Recognition Tasks**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 1, 1999.

[98] R. Plamondon, P. Yergeau, J. J. Brault: **A Multi-level Signature Verification System**. In From Pixels to Features III—Frontiers in Handwriting Recognition, Elsevier Publ., 1992.

[99] J. Psutka: **Komunikace s počítačem mluvenou řečí**, Academia, Praha, 1995.

[100] L. R. Rabiner: **A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition**. Proc. of the IEEE Vol. 77, No. 2, 1989.

[101] L. Rabiner, B.-H. Juang: **Fundamentals of Speech Recognition**, Prentice Hall, Englewood Cliffs, 1993.

[102] N. K. Ratha, S. Chen, A. K. Jain: **Adaptive Flow Orientation Based Texture Extraction in Finger Print Images**. Pattern Recognition, Vol. 28, No. 11, 1995.

[103] N. K. Ratha, A. Senior, R. M. Bolle: **Automated Biometrics**. Lecture Notes in Computer Science 2013 Springer-Verlag, Berlin, Heidelberg, 2001.

[104] D. A. Reynolds: **The Effects of Handset Variability on Speaker Recognition Performance: Experiments on the Switchboard Corpus**. In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996.

[105] S. Rieck, M. Schüßler: **Recognizing Handwritten Address Words with Hidden Markov Models**. Knstliche Intelligenz, Volume 2, Bremen, 1999.

[106] O. Rohlík, V. Matoušek, P. Mautner, J. Kempf: **A New Approach to Signature Verification: Digital Data Acquisition Pen**. Neural Network World, Vol. 11, No. 5, 2001.

[107] O. Rohlík, V. Matoušek, P. Mautner, J. Kempf: **A New Approach to Signature Verification—Digital Data Acquisition Pen**. In Proc. of SoftCOM 2001 Conference, Spilt, 2001.

[108] O. Rohlík, V. Matoušek, P. Mautner, J. Kempf: **HMM Based Handwritten Text Recognition Using Biometrical Data Acquisition Pen**. In Proc. of the CIRA 2003 Conference, Kobe, 2003.

[109] H. Sakoe, S. Chiba: **Dynamic Programming Algorithm Optimization for spoken Word Recognition**. IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 26, No. 1, 1978.

[110] A. Samal, P. Iyengar: **Automatic Recognition and Aanalysis of Human Faces and Facial Expressions: A Survey**. Pattern Recognition, Vol. 25, No. 1, 1992.

[111] L. Schomaker, E. Segers: **Finding Features Used in the Human Reading of Cursive Handwriting**. International Journal on Document Analysis and Recognition, Vol. 2, 1999.

[112] M. Schüßlet, H. Niemann: **Die Verwendung von Kontextmodellen bei der Erkennung handgeschriebener Wörter**. In Proc. of the Musterkennung'97, Brainschweig, 1997.

[113] A. W. Senior: **Off-line Cursive Script Handwriting Recognition Using Recurrent Neural Networks**. Ph.D. thesis, Cambridge, 1994.

[114] C.Y. Suen: **Handwriting Generation, Perception, and Recognition**. Acta Psychologica, Vol. 54, 1983.

[115] C.Y. Suen, J. Guo, Z.C. Li: **Analysis and Recognition of Alphanumeric Handprints by Parts**. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, 1994.

[116] A. F. Syukri, E. Okamoto, M. Mambo: **A User Identification System sing Signature Written with Mouse**. Lecture Notes in Computer Science 1438, Springer-Verlag, Berlin, Heidelberg, 1998.

[117] M. Taft: **Reading and the Mental Lexicon**. Lawrence Earlbaum, New Jersey, 1991.

[118] C. C. Tappert, C. Y. Suen, T. Wakahara: **The State of the Art in On-line Handwriting Recognition**. In IEEE Transactions on Pattern Analysis and Machine Intelligence, No. 8, 1990.

[119] I. Taylor, M. M. Taylor: **The Psychology of Reading**. Academic Press, New York, 1983.

[120] H. K. Tham, R. Palaniappan, P. Raveendran and F. Takeda: **Signature verification System using Pen Pressure for Internet and E-Commerce Application**, ISSRE and Chillarege Corp., 2001.

[121] T.P. Thornton: **Handwriting in America: A Cultural History**. Yale University, 1996.

[122] O. D. Trier: **Goal-Directed Evaluation of Binarization Methods**. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 17, No. 12, 1995.

[123] A. Waibel, T. Hanazawa, G. Hinton, K. Shiano, K. Lang: **Phoneme Recognition Using Time-Delay Neural Networks**. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989.

[124] J. L. Wayman: **Error Rate Equations for the General Biometrics System**. IEEE Automation and Robotics Magazine, Vol. 6, No. 1, 1999.

[125] T. Wessels, C. W. Omlin: **A Hybrid System for Signature Verification**. University of Stellenbosch, 1997.

[126] R. P. Wildes: **Iris Recognition: An Emerging Biometric Technology**. Proc. of IEEE, Vol. 85, No. 9, 1997.

[127] T. K. Worthington, T. J. Chainer, J. D. Williford, S. C. Gundersen: **IBM Dynamic Signature Verification**. In Computer Security, Amsterdam, 1985.

[128] I. Yoshimura, M. Yoshimura: **On-line Signature Verification Incorporating the Direction of Pen Movement: An Experimental Examination of the Effectiveness** In From Pixels to Features III: Frontiers in Handwriting Recognition, Elsevier Publ., 1992.

[129] M. Yoshimura, Y. Kato, S. I. Matsuda, I. Yoshimura: **On-line Signature Verification Incorporating the Direction of Pen Movement**. IEICE Transactions 74, 7, 1991.

[130] A. Zell et al: **SNNS—User Manual**. University of Stuttgart, 1995.

[131] **Question Document Examination**.
http://www.qdewill.com/

[132] **How to Enter Data into a Palm Handheld**.
http://www.palm.com/products/input/

[133] **Pen Computer Technology**.
http://www.fujitsupc.com/

[134] **Non-contact Fingerprint Scanner**
http://www.ddsi-cpc.com/pages/products/cscan300.html

[135] **Markov Modeling of Simple Directional Features for Effective and Efficient Handwriting Verification**
http://www.it.jcu.edu.au/ alan/handwriting/

[136] **The Anoto Techology**
http://www.anoto.com/technology/documents/

[137] **I-Pen producer homepage**
http://www.inductum.com/

[138] **OTM technology overview**
http://www.otmtech.com/

[139] **E-Pen documentation**
http://www.mediaforte.com.sg/products/graphic/epen/

[140] **N-Scribe product documentation**
http://www.n-scribe.com/

[141] **MediPen, SciPen and NotePen documentation**
http://www.compupen.com/

[142] **Cyber SIGN Inc.**
http://www.cybersign.com/

[143] **SignPlus—Intelligent Signature Verification Solutions**
http://www.signplus.com/

[144] **C-Pen Developer Web Page**
http://www.cpen.com/Technology/Developer/

[145] **Interlink Electronics Homepage (ePad)**
http://www.epadlink.com/

[146] **Wintab Driver**
http://www.pointing.com/WINTAB.HTM

[147] **API for Pen and Tablet Drivers**
http://www.penop.com/products/penx/

[148] **iSign—Signature Verification Products by CIC**
http://www.cic.com/products/isign/

[149] **Wacom Tablets Homepage**
http://www.wacom.com/productinfo/

[150] **Microsoft Tablet PC Specification**
http://www.microsoft.com/tabletpc/

[151] **The Origins of Kanji in Japan**
http://www.logoi.com/notes/kanji_origins.html

[152] **NIST Database**
http://www.nist.gov/data/

[153] **Unipen Foundation**
http://hwr.nici.kun.nl/unipen/

[154] **Graffiti and Graffiti 2 Writing Software**
http://www.palm.com/products/input/

[155] **Free On-line Dictionary of Computing**
http://foldoc.doc.ic.ac.uk/foldoc/index.html

[156] **The Design and Implementation of a Stroke Interface Library**
http://www.etla.net/libstroke/libstroke.pdf

[157] **wayV—An Experiment with HCI, Especially Gesture Based Computing**
http://www.stressbunny.com/wayv/

[158] **Gesture Recognition Software**
http://www.handhelds.org/projects/xscribble.html