Adaptive On-line Recognition of Handwriting

Vuokko Vuori

21st August 1998

Contents

Chapter 1

Background

1.1 Introduction

On-line automatic recognition of handwritten text has been an on-going research problem for four decades. It has been gaining more interest lately due to the increasing popularity of hand-held computers, digital notebooks and advanced cellular phones. Traditionally, man-machine communication has been based on keyboard and pointing devices. These methods can be very inconvenient when the machine is only slightly bigger or same size as human palm. A keyboard is very difficult to integrate in small devices and it usually determines the size of the whole apparatus. This is especially true when the number of the characters is very large as in Chinese and Japanese. A pointing device, for example a track ball or a pen, is insufficient or very slowly when used alone and when also textual input is desired.

Because of these problems, new methods for input are developed, for example systems that recognize speech and handwriting. Because both are, very natural ways to communicate, people would easily learn to use to them. Unluckily, these tasks are not that easy for computers whose artificial intelligent is different from that of human's. Both recognition problems are still not completely solved. Very high accuracy is needed for such system before they could be commonly accepted. Handwriting recognition is more attractive input method, especially in noisy environment and when privacy is needed.

Handwriting recognition does not sound a difficult problem. Recognition system should just choose the correct character, usually the one that most reminds the written one, from a limited set of characters. The prominent problem in handwriting recognition is the vast variations in personal writing styles. There are also differences in one person's writing style depending on the context. The writing style may also evolve with time or practice. Also, the mood of the writer and the writing situation can have an effect on style.

A recognition system should be insensitive for minor variations and still be able to distinguish different but some times very similar looking characters. Recognition systems should, at least in the beginning, be able to recognize many writing styles.

Such multi-user systems have usually problems with recognition accuracy. One way to increase performance is adaptation, which means that the systems learns its user's personal writing style.

1.2 Types of handwriting recognition systems

Handwriting recognition systems can be divided to many different categories. Most important discriminating feature of systems is the time when the text is recognized. Recognition can be done either off-line or on-line. Also the different sequences of text which form the units to be recognized as a single entity discriminates problem types: the text can be considered as characters, words or whole sentences. The set of characters, the writing style and the constrains on the writing have also a great influence on recognition methods. These distinctions are described in the following sections in more detail.

1.2.1 Off-line recognition

The handwriting recognition task heavily depends on the application it is used. First applications were systems that recognized text that was originally written on paper. In such systems. paper sheets are digitized to get two dimensional images. Features for recognition are then first enhanged and then extracted from the bitmap image by means of digital image processing. This type of recognition is called off-line recognition as the text is not recognized same time as it is produced but after the writing task is completed. It is a subset of optical character recognition (OCR) (Tappert, Suen, and Wakahara 1990).

Off-line methods are not suitable for man-machine communication because they do not facilitate real-time interactivity. They are suitable for automatic conversion of paper documents to electric documents which then may be interpreted or post-processed by computers. Typical applications of optical character recognition are systems which need to automatically handle huge amount of information in paper form. Such recognition systems have been developed for numerous application areas but are mainly for machine-printed text. Application for handwritten text are, to name a few: automatic sorting of mail, handling of financial documents such as checks, writer recognition, signature verification, and shorthand transcription (Govindan 1990).

1.2.2 On-line recognition

In the case of on-line recognition, text is input with special equipment and it is recognized in real time. Input media is usually a tablet or a flat display that can capture information on the location and the motion of a pen or some other pointing device moving on its surface. Writing can be done with any ordinary pen or a pen-like device specially designed for its purpose. Location and possibly also pressure data are frequently sent to the computer which performs the recognition task.

On-line recognition is considerably different from off-line recognition because of the dynamic information on writing: writing speed, acceleration, angle of the pen and its pressure against the writing surface (Nouboud and Plamondon 1990). Additional information, that is missing in OCR, can be derived from these: the number and the order of the strokes. On the other hand, the dynamic information can be abandoned and methods developed for off-line recognition can be used instead. Just the preprocessing phase must be modified. The dynamic information improves recognition accuracy according to the result of the tests done by E. Mandler, R. Oed and W. Dosters in 1985 (Tappert, Suen, and Wakahara 1990).

The main applications of on-line character recognition allow only a limited interaction as they typically are just for filling out forms. In more advanced applications, the pointing pen and a handwriting recognition system can replace the mouse and the keyboard by emulation. Such systems usually have a special tablet for writing and the recognition is done by the tablet's own microprocessor as the algorithms are very dependent on such things as sampling rate and resolution. Recognition result is sent to the main application whose software only needs minor modifications if any (Tappert, Suen, and Wakahara 1990).

A good example of commercial product is Smart Paper developed at the IBM's Thomas J. Watson Research Center. It is a tablet that can be used with an ordinary PC. The text to be recognized is written on ordinary paper on the top of the tablet, which has a little display for feedback to user and shortcut buttons for commands. Writing is done with a special pen. Tablet's recognition algorithm is adaptive so it can learn the user's writing style (Gardner 1997).

Writing surface can also be integrated to the device as it is usually done in novel hand-held devices called personal digital assistants (PDAs). Newton from Apple, that was released in 1993, was the first PDA without keyboard. It has a touch sensitive screen and the writing is done with natural handwriting style and a special stylus. The system had problems with recognition accuracy mainly because of new users not familiar with handwriting recognition expected the system to recognize also peculiar styles. This problem is avoided in 3Com's Palm Pilot in which writing style is constrained and the system recognizes only special characters called Graffiti. Despite the fact that the user has to learn a new character set, Palm Pilot has dominated the PDA markets for the last two years (Comerford 1998).

1.2.3 Character sets

The character set which form the text to be recognized affects crucially the recognition algorithms used. Recognition task is naturally easier when the number of characters is low and they are dissimilar. Recognition systems have been developed for numerous languages and character sets: Roman alphabets and Arabic numerals, Japanese, Chinese, Indian, Arabic, Korean, Cyrillic, Hebrew, Thai, Greek, Berber, and shorthand writing (Govindan 1990). The most difficult characters are Chinese due to the enormous number of them – over 50 000. Chinese characters can also be written in two different styles, block and cursive.

Other important characteristics of the character sets are the typical number of strokes in one character and the variety of the stroke forms. The Chinese characters have an average of 8-10 strokes in block style. It is considerably more than the average of about 2 strokes in capital Roman alphabets (Tappert, Suen, and Wakahara 1990). In Japanese hiragana, almost all strokes are curved whereas the strokes in the Chinese characters consist mainly of straight lines and only a few corners (Wakahara and Odaka 1997).

1.2.4 Writing style variations

Handwritten characters have enormous variety in shape compared to machine printed characters. Variations occur mostly between-writer but also within-writer. Handwritten characters may be regarded as distorted versions of idealized character models and the distortion can be interpreted to be caused by several factors. According to Kuklinski (1984) sources of variations can be classified as personal background factors, situational factors, and material factors. These factors can affect both the generation and recognition of the characters.

1.2.4.1 Variations of characters

Writing style is based on the alignment and the variating forms of characters. Handwritten characters can vary in both their static and dynamic properties (Tappert, Suen, and Wakahara 1990). Static properties are the underlying, ideal model of the character and the geometrical properties such as relative position and size of the strokes, corners and retraces, ornamentals, aspect ratio, size, and slant. Dynamic properties are more involved with the generative aspects of the characters. Characters can be very similar looking though their number of strokes, stroke order and direction may vary considerably (Kuklinski 1984). Because of these numerous variations the number of styles to write a certain character can be considered as infinite.

The number of strokes varies as sequential strokes have a tendency to be connected within a character. This is caused by pen lift failures. Stroke connecting is more likely to occur between sequential vertical or horizontal strokes whose ends are close than on strokes where one is horizontal and the other is vertical. Retraces of strokes may be made for same reason. Very short retraces in the beginning and the end of the strokes are called hooks and continuations, respectively. The number of strokes increases when a stroke is broken by a false pen lift. The direction of a stroke varies if the writer tries to minimize the time that pen is lifted up or by some personal preferences. Usually, vertical strokes are done before horizontal strokes and strokes on left before strokes on the right (Ward and Kuklinski 1988).

Also, the form of the strokes varies – nominally straight strokes can be curved as bows or s-curves. The forms of loops and cusps chance as loops have tendency to collapse into cusps and, on the contrary, cusps can change to loops. Distinguishing the cusps and loops is difficult problem because they produce similar features (Ward and Kuklinski 1988).

1.2.4.2 Alignment of characters

Handwriting styles can be classified in four basic classes: boxed discrete, spaced discrete, run-on discrete, and connected or pure cursive handwriting. Boxed discrete style means that all characters are written in boxes. Spaced discrete style means that the writing is less constricted and the characters are separated spatially by a significant space or by temporally a predefined time difference between the characters. In run-on discrete style, characters can touch each other but a stroke cannot include more than one character. This means that pen must be, at least, lift up between characters. In the cursive style, all the letters in one word are connected. Hence, the form of the letters are greatly affected by the surrounding letters. Usually, natural handwriting style is a mixture of discrete and cursive styles Bellegarda et al. (1994), Fujisaki et al. (1991).

One of the major problems in the recognition of cursive style handwriting is the segmentation of the words into the characters. A well known paradox states that to attain a good segmentation, the word should be recognized first but before the recognition can be done, the word must be segmented. So called external segmentation methods which are done prior to the recognition can not sufficiently isolate character in case of non constrained cursive handwriting in which several letter can consist of one stroke (Tappert et al. 1990). These methods can be used to find words. Several different segmentations into characters can be then tried for each word and the segmentations yielding the best recognition result in sense of decision criteria are chosen. A lexicon and other linguistic knowledge can be used to prune the alternatives. Segmentation can also be done iteratively on the basis of the recognition results. The segmentation methods based on the recognition are called internal segmentation (Tappert et al. 1990). Sometimes segmentation and recognition are carried out simultaneously in an integrated process. Such a method is described by Hu et al. (1998).

1.2.4.3 Personal background factors

One of the most important personal background factors of writing style is handedness as left-handed and right-handed writers use muscle and tendon groups involved in writing differently. For example, left handers prefer to draw horizontal lines from right to left and right handers prefer to do it in the opposite direction. This can be explained by the fact that writers usually prefer dragging the pen behind the hand to pushing it ahead (Ward and Kuklinski 1988). Age and health affect motor control of writing and thereby the resulting writing style. Also education and origin of the writer have important roles in the writing style because different type characters are taught in schools. In addition some profession affect style, especially when neat writing style is needed (Kuklinski 1984).

1.2.4.4 Situational factors

Situational factors are haste, motivation, and the method of presentation. Haste can be caused by several reasons and it has major effects on style as many people have different styles for different writing speeds. Also, fatigue and mood affect the writing speed. Motivation of writer also determines how carefully characters are written, for example, the address of an important letter is very likely to be written neatly while an ordinary shop list can be almost unreadable even to the writer itself. Text can be written differently whether it is self generated, notes from a telephone conversation or copied (Kuklinski 1984).

1.2.4.5 Material factors

The writing instrument and surface and writing form constitute the material factors of writing style. The writing instrument has an effect on the writing style depending on its size and overall comfortableness. Writing surface's friction and position also affect style. The form factors, such as the size of the blank writing area, the length of writing line, or the size of the writing boxes for characters, can have a dramatic effect on the style (Kuklinski 1984).

1.2.4.6 Constraints on writing

As the variations are the key problem of handwriting recognition, writing style is usually more or less constrained. Usually, characters are written in boxes to ease their separation, or the system provides guiding lines to help the users to write more consistently. Constraints are designed to improve the performance of the recognizer. The existence of the constraints is mediated to the recognizer as a set of a priori rules. If theconstrains are too strict, they may cause the system to be rejected by the users. According to tests done by Suen (1979) constrains do affect the writing style, the writing speed and the recognition results.

In the most constrained systems, users are expected to use specific characters. These systems perform well, assuming that the user writes in the required style. The drawback of such a system is that the users tend to write in their own style and, as the styles vary a lot, the system is awkward for most of the users (Ward and Blesser 1985).

Esimerkki, kuva Graffitimerkistosta?

1.2.5 Context

Text is a string of characters which can be divided into basic units such as syllables, words, phrases, sentences, lines, paragraphs, and so on. Context of the text are the surrounding units which give meaning for the text. This context can be used to a help in recognizing text and, especially in handling confusing parts, to avoid or reduce faulty recognitions. According Srihari (1985), contextual knowledge includes the following: the probabilities of letters, letter pairs, letter triplets etc.; probabilities of words; legal combinations of letter pairs, triplets, etc.; a lexicon or list of words acceptable in global or local context; a grammar describing the syntax of the language; and a semantic network representation of the concepts embodied.

The context is very important in recognition of cursive script, especially during the character segmentation phase. Contextual information can also be used when recognizing discrete characters, and it usually increases the system's performance (Tappert, Suen, and Wakahara 1990). In some problems, no contextual information exists. The language used can be unknown and the input text totally unconstrained so that the text may not have any apparent meaning. Situation like this is very likely to occur in a PDA application in which the user typically writes down names and nicknames, addresses, telephone numbers, codes and other short notes. It is not possible to know what language is used, when it is changed nor make a dictionary with all existing words in it.

1.3 Benefits of pen based input

The most important benefit of pen based input is combined pointing and text input. An application with graphical user interface requires the user to switch between keyboard and pointing device if the user wants to manipulate icons, menus, windows, etc, and to enter text in different locations. Switching between a keyboard and a pointing device means in practice that the user has to move his/her hand between the input devices and re-establish visual frame of reference among the display, the keyboard, and the pointing device.

When both the pointing information and text input are used simultaneously, nonlinear text input is possible. Nonlinear text input means writing in which characters, words and sequences are not written in the same order as they are read (Ward and Blesser 1985).

Compared with a small keyboard, writing with pen is easier and not slower than typing. With a miniature keyboard integrated in small device, typing speed is 10-21 words per minute, and with a pen 15-18 wpm. Both speeds are considerably less than typing speeds for touch screen keyboard and ordinary keyboard, 20-32 wpm and about 60 wpm, respectively (Guyon and Warwick 1996), (Ward and Blesser 1985).

Compared to text input based on speech recognition handwriting offers better privacy. Handwriting is not sensitive to noise and it does not cause disturbance in environments where silence is required, for example in a library. In addition, the user does not need to spell letter by letter difficult words, such as foreign words and names, or remember long and complicated sentences while dictating text to the system. Also, editing of the text afterwards is easier (Mel et al. 1988).

1.4 Requirements for recognizer performance

Recognition accuracy is the key factor in determining the acceptability of a handwriting recognition system and the whole application where it is implemented. According to tests done by Frankish, Hull, and Morgan (1995), the relationship between the user acceptance of pen interface and the recognition accuracy is highly task-dependent.

Guyon and Warwick (1996) have made tests where good typists wrote with a special keyboard that made random errors with a predefined rate. These tests showed that writers tolerate errors up to 1% while 0.5% is unnoticeable and 2% is intolerable. These error rates are lower than error rates for human readers. Test done by Neisser and Weene (1960), which involved 200 writers and 9 readers, showed that average human recognition error rate for isolated, handwritten, and context free characters is 4.1%. Almost the same result (4.0%) was obtained in a more recent test where context free characters from 10 writers was recognized by 10 readers. The best writer-reader couples had zero error percentages but the error rate for the worst couple was as high as 21.7% (Parizeau and Plamondon 1994).

These tests clearly show that the required recognition rate for handwriting recognizer is very high. Another requirement for the recognition systems, besides the recognition rate, is the reliability. Reliability means that system rather rejects characters than recognizes them wrongly.

1.5 Recognition methods

The character recognition problem has been tried to solve with many different methods. Some of the methods are especially developed for character recognition but most of them are loaned from other fields of pattern recognition, signal processing, and image analysis. Recognition methods can roughly be classified in three major groups: statistical, structural and syntactical, and model matching methods. Sometimes, different methods are combined, for example, simple methods are used for preclassification and final decision is done with more sophisticated methods.

The recognition procedure is basically very simple: after preprocessing, some features are extracted from the unknown character which is then classified to the class whose members have the most similar features. The problems are: 1) finding both descriptive and discriminating features, 2) selecting a way to compare them, and 3) creating the classification rules. Of course, these problems are highly dependent on each other.

In the next sections, some character recognition methods are briefly described. In the recent years, the number of commercial applications of handwriting recognition is increased. Due to the hard competition for market shares, research is not so open as it used to be, and it is difficult to find information on the most recent methods.

1.5.1 Statistical methods

In statistical methods, the recognition of a character is carried out by choosing the character class which is most probable or has minimum measure of expected classification error or some other risk. Features are assumed to be variables with statistical distributions, and different classes to have different values for the distribution parameters. The values of the parameters can be estimated from measurements. If the parameters are updated on-line, the systems is adaptive and thus able to adjust to different writing styles. Also, *a priori* class probabilities can be used in the decision rule.

In following, some methods are described in more detail as examples of statistical methods. The recognition rates of different methods are not as such comparable as the methods have been created and tested with different data.

Example 1: A simple statistical method for on-line character recognition is described by Odaka et al. (1982). Isolated characters are preclassified according to the number of strokes. Each stroke is then approximated with a small number, N, of points which divide the stroke to straight segments of equal lenght. The reference patterns consist of the mean vectors and covariance matrices of the strokes. Final classification is done by minimizing the distance between the feature vector of the unknown character and the reference patterns. The feature vector of a character consisting of S strokes is:

$$L = (x_{11}, y_{11}, x_{12}, y_{12}, \ldots, x_{SN}, y_{SN}). \tag{1.1}$$

The distance between a feature vector L and a reference pattern of class θ is defined as follows.

$$D(L, \theta) = (L - M_\theta) {\textstyle\sum_\theta^{-1}} (L - M_\theta)^T, \tag{1.2}$$

where M_θ is the mean vector and $\sum_\theta$ is the covariance matrix of the reference patterns.

It was found that the suitable number of points per strokes is five for alphanumerics and over 100 training samples per class are needed. The mean vectors are the covariance matrices are estimated for each class using the corresponding training samples. The system was trained with data from many writers and a surprisingly high (99.9%) recognition rate for alphanumerics was achieved.

Example 2: Method described in the previous example is more suitable for characters that consist of straight strokes as curly strokes can not be estimated properly with few points. Another on-line method for more curly characters is proposed by Arakawa (1983). In that method the feature vector of a stroke consists of P Fourier coeffients for pen-point movement loci. The feature vector of a character of S strokes is:

$$L_s = (a_{11}, \ldots, a_{1P}, b_{11}, \ldots, b_{1P}, \ldots, a_{S1}, \ldots, a_{SP}, b_{S1}, \ldots, b_{SP}), \qquad (1.3)$$

where

$$a_{ij} = \frac{2}{N_i} \sum_{j=1}^{N_i} x_{ij} \cos(\pi ij/N_i) \qquad (1.4)$$

and

$$b_{ij} = \frac{2}{N_i} \sum_{j=1}^{N_i} y_{ij} \cos(\pi ij/N_i). \qquad (1.5)$$

(x_{ij}, y_{ij}) is the jth point in ith stroke. N_i is the total number of points in the ith stoke.

The method is otherwise very similar to the method described in previous example. Classification is based on the Bayesian decision rule: the distributions of the pattern feature vectors are assumed to be multivariate normal and the most probable class is chosen. The probability of a feature vector L belonging to class θ is:

$$P(L, \theta) = \frac{1}{(2\pi)^{n/2} |\sum_\theta|^1/2} \times \exp\left[-\frac{1}{2}(L - M_\theta){\sum}_\theta^{-1}(L - M_\theta)^T\right], \qquad (1.6)$$

where M_θ is the mean vector and $\sum_\theta$ is the covariance matrix of reference pattern in the class θ. n is the length of the feature vector and $n = P \cdot S$.

With this method high recognition rate (99.2%) for alphanumerics and multiple writer data was obtained.

1.5.1.1 Chain codes

Different chain codes are widely used in character recognition methods. Chain coding means that the character is divided into segments which are then coded. The segmentation can be done by local extreme points (left, right, up, down, maximum or minimum velocity etc.) or by evenly spaced points. Codes can present straight, equally length lines with discrete directions, lengths of straight lines and continuous angles between them, or different types of lines, curves, loops and corners.

Classification of chain coded characters can be done in several ways, for example statistical methods (Loy and Landay 1982), string comparison methods (Nouboud and Plamondon 1991) or dynamic programming (Yuen 1996) can be used.

Example 3: A statistical method to classify chain coded characters is described by Loy and Landay (1982). In this method segmentation is done according to the maximum curvature points and a character is approximated with a polygon. A feature vector consists of the distances of the line segments from the origin, and the angles between the line segments and the horizontal axis. The pattern feature vectors are assumed to have similar distribution as in the previous example. Also similar classification rules is used as in both previous examples.

The method was tested with data from single writer. The recognition rates for both classification rules were over 98% and when the distribution parameters were up-dated on-line, the recognition rates improved up to 99%.

1.5.1.2 Hidden Markov Models

The speech recognition problem is quite similar to the handwriting recognition problem and so many methods and ideas are adopted from the former problem to the latter. Hidden Markov models (HMMs) are currently the state of the art -method in speech recognition, and they seem to be very promising and gaining popularity also in handwriting recognition. They are especially suitable for the recognition of cursive script.

HMMs were initially introduced and studied in the late 1960s and early 1970s. HHMs are signal models, and they are also called Markov sources and probabilistic functions of Markov chains. They are used to describe the statistical properties of the signal. The modeled signal is assumed to be a random process whose parameters can be estimated in precise, well-defined manner. HMMs are based on Markov processes which states are not observable – that is why they are called hidden Markov models. Observations are probabilistic functions of the states.

A basic HMM is characterized by the following parameters: the number of states in the model (N),the number of distinct observations per state (M), the state transition probabilities ($A = \{a_{ij}\}$, where i and j are the indexes of the corresponding states), the observation probabilities for each state ($B = \{b_j(k)\}$, where j is the index of the state and k is the index of the observation), and the initial state distribution ($\pi = \{\pi_j\}$, where j is the index of the state) (Rabiner 1989). HMMs used in handwriting recognition are typically so called left-right models (Hu, Brown, and Turin 1996). These models have the property that the state index increases or remains unchanged as the time increase. A direct path (the state index increases by one in every state transition) through such a model corresponds to the average realization of the modeled text element. Self loops (the state index remain unchanged) correspond to the lengthened versions of the element and state skips correspond to the shortened versions (Nathan, Bellegarda, Nahamoo, and Bellegarda 1993).

An HMM can represent several things in a handwriting recognition system. Early HMM-based on-line recognition systems used separate models for each word. Because the number models increases with the size of the vocabulary, systems using character models have become more popular (Nathan, Bellegarda, Nahamoo, and Bellegarda 1993), (Veltman and Prasad 1994). As the characters usually contains patterns that are common to more than one character, also stroke and other subcharacter models can be used. They can be concated to form letter models (Hu, Brown, and Turin 1996). The training of the system is easier when the number of models is low. In case of a large vocabulary it can be difficult to obtain adequate training set as several samples for each word model is needed so that the model parameters could be estimated properly.

In the speech recognition systems, the observation probabilities for each state are usually mixtures of different continuous probability densities ($b_j(O) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(O, \mu_{jm}, U_{jm})$, where O is an observation vector, c_{jm} is the mixture coefficient for the mth distribution in the jth state, and $\mathcal{N}$ is any log-concave or elliptically symmetric probability density function). Discrete probability densities are more suitable for handwriting recognition, according to the test done by Rigoll et al. (1996). Similar result were obtained by Bellegarda et al. (1995). In a writer-dependent task the recognition results of the discrete parameter method were as good as the results of the continuous parameter method, and in a writer-independent task the results of the discrete parameter method were only slightly worse. In both tasks the discrete parameter method required only a few percent of the CPU time consumed by the continuous parameter method.

Three basic problems associated with HMMs are computing the probability of the observation sequence for a given model, choosing a corresponding state sequence for a given observation sequence and model, and adjusting the model's probability parameters A, B and π according to a given observation. The fist problem is the same as choosing the model which best matches the observation. Solving the second problem corresponds to uncovering the hidden parts of the model. The solution of the last problem is a method to train and adapt the system (Rabiner 1989).

Example 4: To demonstrate the similarities between the problems and the suitability of the method, Starner et al. (1994) used an HMM-based system developed for speech recognition to recognize handwritten text. Only modification of the system was the change of the front end module – different features were used for handwriting than speech. The system could recognize script written on-line. Features used were the writing angle and its change in each sampled point. Both features were quantized. A 7-state HMM were used to model each character.

The results were good:an average of 4.2% word error rate and an average of 1.4% character error rate was achieved for the writer dependent tasks. When a statistical bigram grammar for word pairs and additional HMMs for the contextual effects of adjacent characters were used, the recognition performance of the system improved and the word error rate dropped to 1.1%.

Example 5: In an article written by Bellegarda et al. (1994), the mathematical background of a recognition algorithm based on HMMs is well described. The algorithm is suitable for unconstrained handwriting. Its main goal is to retrieve the actual character sequence $\vec{C}$ from the handwritten sequence $\vec{S}$. In statistical sence this means finding the character sequence $\vec{C}^*$ that minimizes the recognition error probability, or in other words, maximizes the *a posteriori* probability for given input sequence $\vec{S}$. The decision criteria of the algorithm can be formulated in the following way:

$$\vec{C}^* = \arg\max_{\vec{C}} Pr(\vec{C}|\vec{S}). \tag{1.7}$$

When Bayes' rule is used and the irrelevant denominator $Pr(\vec{S})$ is omitted:

$$\vec{C}^* = \arg\max_{\vec{C}} Pr(\vec{S}|\vec{C})Pr(\vec{C}), \tag{1.8}$$

where the first term $Pr(\vec{S}|\vec{C})$, which can be called handwriting channel, characterizes all the scribble sequences that can be generated for the intended character sequence $\vec{C}$, and the last term, the source model, $Pr(\vec{C})$ characterizes the possible sequences of characters in the language considered.

The input sequence $\vec{S}$ consists of frames, which are small uniquely informative portions of handwriting. A character sequence is composed of characters which can be associated with a block of frames delimited by two boundary points. In case of discrete handwriting, the boundary points can be uniquely determined and the handwriting channel can defined in the following way:

$$Pr(\vec{S}|\vec{C}) = \prod_{j=1}^{M} Pr(\Phi_j|c_j), \tag{1.9}$$

where $\Phi_j = f_{b_{j-1}+1} \ldots f_{b_j}$ is a block of frames associated with the character c_j. $\{b_j\}$ are the indexes of the bounding points. When writing is connected, the boundary points are quite arbitrary and the handwriting channel is defined differently:

$$Pr(\vec{S}|\vec{C}) = \sum_{\{b_l\}} \prod_{j=1}^{M} Pr(\Phi_j|c_j), \tag{1.10}$$

where the summation is taken over all possible choices of the boundary points, which segment $\vec{S}$ to M characters. $Pr(\Phi_j|c_j)$ is defined with an HMM, which is so called baseform $\mathcal{B}_j$ of character c_j. It corresponds to the probability that the block Φ_j is a realization of given baseform $\mathcal{B}_j$. The definition of the probability is:

$$Pr_{\mathcal{B}_j}(\Phi_j) = \sum_{\{paths\}} Pr(\Phi_j|path)Pr(path), \tag{1.11}$$

where $\{paths\}$ are all possible routes trough the associated HMM. Both probabilities needed are defined by the same HMM.

1.5.2 Model matching

The model matching recognition methods are based on an assumption that the handwritten characters (or other text units such as whole words or strokes) are distorted realizations of ideal models. Distortions are explained by the handwriting style variations, changes in the orientation and scaling of writing, and the noise from the environment or the equipment. In OCR model matching means comparison between two images but in on-line recognition it usually means curve matching. The curves matched can consist of different functions of time, for example the preprocessed x and y coordinates of the pen point, the pressure against the writing surface and the angle of the trajectory. Curve matching is either linear or nonlinear. In the former case, the number of data points is constant and one-to-one correspondance can easily be defined between the points of two curves, where as in the latter case, this is not true.

An unknown pattern is recognized by finding the model which minimizes the distance between the two patterns or maximizes the similarity measure. Many different distances and similarity measures have been suggested and successfully used. In the following examples, some of them are described in more detail.

The recognition systems based on Neural Networks (NN) use a similar basic idea as model matching methods. Instead of explicitly storing models in a model base, the information of models is stored implicitly in the weights of the neurons. The input of NN consists of the features extracted from the patterns to be recognized, and the output is the scoring values for each class or simply the number of best matching class. NN are widely used for various pattern recognition tasks. Some NN-based applications and methods for handwriting recognition are described by Guyon and Wang (1993). NNs are so called black box models and thus it is difficult to analyze their recognition process and increase the recognition performance by finding and elimating the error sources. They also need considerably large training data bases. HMMs have similar drawbacks compared to model matching.

Example 6: Wilfonng et al. (1996) have developed an on-line recognition method which is based on linear curve matching. It is suitable to an user specified alphabet, which can consist of all kinds of handwritten text units or drawn symbols, for example characters, words, or any arbitrary pictures and symbols. These handproduced patterns are called symbol instances. Depending on its use, a symbol instance can be called model instance or unknown instance: the model instances form the alphabet and the unknown instances are recognized by them. Instance symbol α is represented by an instance vector

$$\vec{v}_\alpha = (x_1(\alpha), y_1(\alpha), \ldots, x_n(\alpha), y_n(\alpha)), \tag{1.12}$$

where $x_i(\alpha)$ and $y_i(\alpha)$ are preprocessed coordinates of the ith data point and n is the total number of the sampled data points. Preprocessing is done by subtracting the coordinates of the centroid of the instance from the original coordinates of the data point. The centroid is the mass center of a piecewise linear curve, which is formed by interpolating line segments between consecutive data points. It is defined in the following way:

$$x_\alpha = \frac{\sum_{i=1}^{n-1} d_i(x_i(\alpha) + x_{i+1}(\alpha))}{2L} \tag{1.13}$$

$$y_\alpha = \frac{\sum_{i=1}^{n-1} d_i(y_i(\alpha) + y_{i+1}(\alpha))}{2L}, \tag{1.14}$$

where d_i is the Euclidean distance between the ith and the $i+1$th data point and L is the sum over all d_is.

Two instances are regarded as equivalent if they can be derived from each other by scaling and rotation. A two dimensional hyper plane trough the origin of an instance space can be constructed for every instance. It presents all the equivalent instances which can be derived from the given instance. The distance between two instances is defined by the angle between the corresponding hyper planes. When the angle is zero, instances are equivalent. The hyper plane S_α consisting of all the instances equivalent to α is

$$S_\alpha = \{w_\alpha(\gamma, r) : 0 \leq r \leq \infty \cap -\pi < r \leq \pi\}, \tag{1.15}$$

where

$$\vec{w}_\alpha(\gamma, r) = r(\vec{v}_\alpha \cos(\gamma) + \vec{u}_\alpha \sin(\gamma)), \tag{1.16}$$

$$\vec{u}_\alpha = (-y_1(\alpha), x_1(\alpha), \ldots, -y_n(\alpha), x_n(\alpha)), \tag{1.17}$$

γ is the rotation angle, and r is the scaling factor.

The recognition is based on minimizing the distance between the unknown instance and the model instance. Before the distance can be calculated, both instances must be modified so that they have same number of data points and strokes. That is done by merging and splitting the strokes of the unknown instance and resampling the piecewise linear curves representing the stokes of the model instance. After stroke merging and splitting, the relative arc lengths of the corresponding strokes of the unknown instance and the model instance are similar or the differences are in given tolerance. The relative arc length of the stoke is obtained by dividing the original arc length by the sum of the arc lengths of all stokes (L). Resampling is done so that the normalized distance between adjacent points is similar to the corresponding normalized distance of unknown instance. In this case, the normalized distance is the Euclidean distance between the points divided by the total arc length of the stroke.

The distance between the unknown instance α and the model instance k is given by

$$\frac{(\vec{v}_\alpha \cdot \vec{v}_k)^2 + (\vec{v}_\alpha \cdot \vec{u}_k)^2}{(\vec{v}_k \cdot \vec{v}_k)^2}, \tag{1.18}$$

when a redundant term $(\vec{v}_\alpha \cdot \vec{v}_\alpha)^2$ is omitted. This distance is insensible to the distortion caused by translation, rotation and scaling. In case of confusing alphabet pairs which differ only by size or rotation, also the rotation angle γ and the scaling factor r can be used for discrimination. They are given by

$$r = \frac{\sqrt{(\vec{v}_\alpha \cdot \vec{v}_k)^2 + (\vec{v}_\alpha \cdot \vec{u}_k)^2}}{(\vec{v}_\alpha \cdot \vec{v}_\alpha)} \tag{1.19}$$

and

$$\gamma = \arccos\left(\frac{(\vec{v}_\alpha \cdot \vec{v}_k)}{\sqrt{(\vec{v}_\alpha \cdot \vec{v}_k)^2 + (\vec{v}_\alpha \cdot \vec{u}_k)^2}}\right). \tag{1.20}$$

This method was tested for capital letters from A to Z. Two samples of each character were collected from eight subjects. One set of characters was used as a model base and the other set were used as test base. The average recognition percentage for writer-dependent tasks was 98%.

Example 7: Wakahara and Odaka (1997) have developed a linear curve matching method for isolated characters which uses a distance completely different from the one used in the matching method described in the previous example. The variations between the unknown character and the model character are explained by the stroke-based affine transformation (SAT) components and the residual components. The SAT components are caused by affine transformations, such as translation, rotation, scaling and shearing, and they represent innerclass variations, where as the residual components represent interclass variations. The matching method consist of three stages: the deformation extraction, the deformation analysis, and the deformation estimation stage.

In the deformation extraction stage, the deformation vector field (DVF) between the unknown character $T = (\vec{t}_1, \ldots, \vec{t}_K)$ and the model character $R = (\vec{t}_1, \ldots, \vec{t}_K)$ is defined. $\vec{r}_k$ and $\vec{t}_k$ are vectors consisting of the coordinates of the kth data points. K is the total number of the data points. Before The DVF can be calculated, the model character has to be modified by stroke linkage method so that it has same number of strokes (N) as the unknown character. After that, both characters are resampled so that the number of data points per stroke (n) is constant and the points are equidistant. The DVF is a set of deformation vectors $\{\vec{d}_k\}_{k=1}^{K}$ given by

$$\vec{d}_k = \vec{r}_k - \vec{t}_k. \tag{1.21}$$

In the deformation analysis stage, the DVF is decomposed into SAT components and residual components. First, the optimal affine transformation is found for each corresponding stroke pair. This is done by solving the following problem:

$$\min_{A_i, \vec{b}_i} \sum_{k \in \{k_i\}} \|(A_i \vec{t}_k + \vec{b}_i) - \vec{r}_k\|^2, \tag{1.22}$$

where i is the index of the stroke pair, A_i is a 2×2 matrix representing rotation, scaling and shearing, $\vec{b}_i$ is a translation vector, and $\|\cdot\|$ is the Euclidean distance. $\{k_i\}_{i=1}^{n}$ is a set of data point indexes of the ith stroke and its L neighboring strokes, which are included because the adjacent strokes affect each other. After A_i and $\vec{b}_i$ are found, the decomposition of DVF is done in the following way:

$$\begin{aligned} \vec{d}_{i,j} &= [(A_i \vec{t}_{i,j} + \vec{b}_i) - \vec{t}_{i,j}] + [\vec{r}_{i,j} - (A_i \vec{t}_{i,j} + \vec{b}_i)] \\ &= \langle \vec{d}_{i,j} \rangle + \vec{\varepsilon}_{i,j}, \end{aligned} \tag{1.23}$$

where i is the index of the stroke, j is the index of the data point, $\langle \vec{d}_{i,j} \rangle$ is the SAT component, and $\varepsilon_{i,j}$ is the residual component of $\vec{d}_{i,j}$.

In the deformation estimation stage, two distances are calculated: D_1 (distance between T and R) and D_2 (distance between SAT-superimposed T and R). D_1 is

$$D_1 = \frac{1}{N} \sum_{i=1}^{N} \gamma_i \sum_{j=1}^{n} |\vec{d}_{i,j}|/n \tag{1.24}$$

and D_2 is

$$D_2 = \frac{1}{N} \sum_{i=1}^{N} \gamma_i \sum_{j+1}^{n} |\vec{\varepsilon}_{i,j}|/n. \tag{1.25}$$

The value of γ_i depends on the modifications done to R by the stoke linkage method: if the ith stroke of R is artificially connected, $\gamma_i \leq 1$, otherwise $\gamma_i = 1$. $|\cdot|$ is the city block norm.

The recognition is done hierarchically in two stages. In the first stage, the distances (D_1) between T and all the model characters are calculated and the nearest 20 models are selected as candidates. In the second stage, the distances (D_2) between T and the candidate models are calculated. The class of the nearest candidate determines the recognition result.

1.5.2.1 Elastic matching

Elastic matching, which is also called dynamic time warping (DTW), is a nonlinear matching method originally used in speech recognition. It was developed in the beginning of 1970s and was introduced as an handwriting recognition method in late 1970s. Similar methods have also been used successfully in on-line signature verification (Martens and Claesen 1997), (Sato and Kogure 1982), (Wirtz 1998).

Dynamic time warping can be used in comparison of all kinds of continuous functions of continuous parameter, for example time. It is based on idea that the speed of the process underlying the function can variate. The effect of these variations can be omitted in the comparison of two functions if the parameter axis of functions are warped. In this case warping means compressing and stretching of the parameter axis locally. In practice, the continuos functions are converted into discrete sequences by sampling the function with regular intervals, or by some other segmentation method based on the character of the function (Sankoff and Kruskal 1983).

The time variation are especially important in signature verification. A forger can write a very similar looking signature, at least after practice, but can not imitate the correct writing speed and pressure because of the difficulties in observing them and the differences in muscular systems involved in writing and their dynamics (Sato and Kogure 1982). Contrary to the signature verification methods in which time warping is used to extract and measure the variations of the generation of similar looking writing styles, in the handwriting recognition methods it is used to minimize these variations. In the signature verification, whole signature is usually handled as unit and no segmentation to smaller units, such as word and characters, is done.

Elastic matching methods used for handwriting recognition are insensible to the distortion caused by different writing speed, for example the variations of the total number and the distribution of the sample points in the character. Typically writing speed slows down in sharp corners, at the other points of extremal curvature, in the beginning and the end of the strokes, but also by hesitation and pausing of the writer. When writing is done slowly, the sample point are located densely, where as quick writing produces more sparsely located sample points. Writing speed variations also cause distortions in the shapes of the strokes. In the elastic matching methods, the optimal time warping and point-to-point correspondence is found by dynamic programming methods which minimize the shape dissimilarities. In handwriting recognition matching can be done on stroke, character or word basis.

Even though elastic matching can cope most of the time variations, it is sensible to the stroke order which can not be changed only by stretching and compressing time. The variations of the number of the strokes can be handled in a similar way as in linear curve matching methods by preprocessing methods which, for example, connect the accidentally broken strokes, detect and remove the retraces and pen lift failures, or connect all the strokes to a sigle stroke in some systematic way.

Since both elastic matching and HMMs are one of the state-of-the-art handwriting recognition methods widely used, Bellegarda et al. (1994) did some tests to compare them. Two systems, based on elastic matching and HMM, were trained and tested with same data. The computational complexities of the systems were approximately same. Both the recognition performaces and CPU times consumed were compared. The HMM system had a significantly lower average recognition error rate for both the writer-dependent and the writer-independent tasks. However, the elastic matching system performed better for some of the writers. The CPU time consumed by the HMM system compared to the time used by the elastic matching system was 50% more in writer-dependent tasks and 30% less in the writer-independent tasks. The choice between the elastic matching methods and the HMM methods can not be done only on the basis of these results: the handwriting recognition problems are highly task dependent, and the performance of the methods can vary considerably depending of the choice of the features, the definition of distance, and the data used in training and testing of the system.

In the following examples, some methods based on elastic matching are described in more detail.

Example 9: A very basic elastic matching method is introduced and described by Tappert (1984). The method consists of three modes of operation: training, recognition, and updating. In the training mode, the prototype set is established by storing the parameters of the characters which the user writes according to a specific training text prompted by the system. In the recognition mode, the unknown characters are classified to the class corresponding to the best matching, or closest, prototype. The prototype set is modified in the updating mode, in which the user writes characters in a similar way as in the learning mode. The characters which are misrecognized are added to the prototype set. Also the characters whose ratio of the distances to the second best and the best matching prototype is less than given tolerance are added to the prototype set.

Due to the updating mode, the recognition systems is adaptive. Before its use the system can be trained with data from several writers, and it can be updated with data from the specific user when problems in the recognition occur. Another suitable method for adaptation of prototype set, which does not in crease the number of prototypes, is proposed by Li and Yeung (1997). In this method, all characters successfully recognized are added to the prototype set whereas the prototype belonging to the same class and corresponding to the lowest value of so called prototype evaluation function is removed. The aim of this method is to produce a prototype set with small intraclass distance and large interclass distance.

The recognition of the handwriting is done on character basis. The characters are separated spatially by the writer. Before the feature extraction the characters are preprocessed: the hooks are eliminated, the data corresponding to a dot are reduced as a single point, and the data points are resampled so that they are roughly equally spaced in distance instead of time. The feature vector of the character consist of the slope angle ϕ_i of the curve, the height y_i, and the normalized coordinates $\overline{x}_i$ and $\overline{y}_i$ from the mass center of the character corresponding to each data point. The distance between two data points, $P_i = (\phi_i, y_i, \overline{x}_i, \overline{y}_i)$ and $P_j = (\phi_j, y_j, \overline{x}_j, \overline{y}_j)$, is given by the equation:

$$d(i,j) = \min\{|\phi_i - \phi_j|, 360^\circ - |\phi_i - \phi_j|\} + |y_i - y_j| + |\overline{x}_i - \overline{x}_j| + |\overline{y}_i - \overline{y}_j|. \tag{1.26}$$

The distance between the unknown character and the prototype character is found by minimizing the following equation:

$$D = \min_{\{w(i)\}} \sum_{i=1}^{N} d(i, w(i)), \tag{1.27}$$

where $\{w(i)\}$ is a set of all possible time warping functions defining the point-to-point correspondence between the characters, and N is the number of data points of unknown character. The time warping function is constrained by boundary condition and continuity condition.

In this case, boundary condition ensures that the first and the last data points of unknown and prototype are matched against each other, respectively. The continuity condition restricts the matching of the other points: the warping function is a monotonically increasing function, multiple data points of the prototype can be matched to a single point of unknown where as only one point of unknown can be matched to a single point of the prototype, and all the points of the unknown are matched but stray points of the prototype can be skipped. This kind of restricted time warping is also called band-limited time waping. In the method described by Li and Yeung (1997) less strict boundary condition and symmetric continuation condition are used. So called virtual sink and source points are augmented both to the unknown and the prototype. The distances between the sink and the source points are set to zero, respectively. This allows the omission of the points in the end and the beginning of the curve.

The distance can be solved recursively by dynamic programming in the following way:

$$D(i,j) = d(i,j) + \min\{D(i-1,j), D(i-1,j-1), D(i-1,j-2)\}, \tag{1.28}$$

where $D(i,j)$ is the cumulative distance to the point (i,j). In the beginning of the recursion, $D(1,1)$ is set to $d(1,1)$ and $D(i,j)$ is set to the infinite elsewhere. All the possible paths from the point (i,j) to the point (N,M) (M is the number of the points of the prototype) are examined recursively, and the path producing the minimal distance is chosen to present the time warp function.

In her master thesis Scattolin (1995) has proposed a modified version of the elastic matching method described above. The optimal dynamic time warping function $w(i)$ is found in the exactly similar way. The main difference between the methods is the definition of the distance between the characters: D is a weighted sum of the point distances and it is given by the equation:

$$D = \sum_{i=1}^{N} \omega_i d(i, w(i)), \tag{1.29}$$

where the weights ω_i are calculated separately for each point of the prototype, and $d(i,j)$ is the squared Euclidean distance of the normalized coordinate points from the mass center of the character. The values of the weights should be high if the corresponding points are important discriminators between two classes. They are calculated according to the recognition results of the training set. Next, a slightly modified version of her method is described. Weights can be calculated in the following fay:

$$\omega_i = \frac{\overline{dtot}(i,E)}{\overline{dtot}(i,C)}, \tag{1.30}$$

where E is subset of the training set and consist of the characters which are misrecognized by the given prototype, whereas subset C consist of correctly recognized characters, and $\overline{dtot}(i,S)$ is the average distance of the points of characters belonging to the set S and matched to the ith point of the prototype. $\overline{dtot}(i,S)$ is given by the equation:

$$\overline{dtot}(i,S) = \frac{1}{size(S)} \sum_{\forall k \in S} d(i, w^k(i)), \tag{1.31}$$

where $w^k(t)$ is the optimal time warp function corresponding to the character k. The weights can be calculate iteratively, for example in the following way:

$$\omega_i^{n+1} = \sqrt{\omega_i^n * \frac{\overline{dtot}(i,E)}{\overline{dtot}(i,C)}}, \tag{1.32}$$

where n is the counter of the iteration rounds, and E and C correspond to the recognition results of the training set when the old weights are used. The initial values of the weights can be set to 1.

The recognition performance of the basic method proposed by Tappert (1984) was tested for Latin characters, both upper and lower case, and Arabic numbers. The average recognition result for six writer-dependent tasks was 94% when prototype set contained all character groups, 98% when separate prototype sets were used for each character group, and nearly 100% when a dictionary was used. The weighted method was tested for Arabic numbers and multi-writer data. The use of the weights increased the recognition rate of the writer-independent task from 85.22% to 88.67%.

Example 10: A physics based shape metamorphosis method for nonlinear curve matching and handwriting recognition has been proposed by Pavlidis et al. (1997). The method, originally used in computer based animation, was developed by Sederberg and Greenwood (1992). The basic idea of the method is that similar shapes undergo less extensive metamorphosis than different shapes when they are morphed to each other. The degree of morphing can be used as matching criteria. The method can be used for the matching of all kinds of handproduced shapes.

A curve is modeled by its perceptually important points, the corner points, and by the points of low curvature between them. These points are found by the algorithm originally proposed by Brault and Plamondon (1993). The linear segments between the points are considered as a virtual wires made of some idealized metal. The shape morphing is done by stretching and bending these wires. The degree of morphing corresponds to the minimal energy needed in the morphing. The minimal energy, and the point-to-point correspondance as by-product, is found by a dynamic programming algorith similar to the one described in the previous example (Sederberg and Greenwood 1992).

The energy needed to stretch a wire segment is given by the equation:

$$E_s(L_I, L_T) = f_s \frac{|L_T - L_I|^2}{(1 - c_s)\min(L_I, L_T) + c_s \max(L_I, L_T)} \tag{1.33}$$

where L_I is the initial length, L_T is the target length, f_s is the stretching stiffness parameter, and c_s is the penalty parameter for segments collapsing to point during metamorphosis.

The bending energy needed to change the angle ϕ between two wire segments is given by the equation:

$$E_b(\phi) = \begin{cases} f_b(\Delta\phi + m_b\Delta\phi^*)^2 + p_b, & \text{if } \phi \text{ does go to zero,} \\ f_b(\Delta\phi + m_b\Delta\phi^*)^2, & \text{if } \phi \text{ never goes zero,} \end{cases} \tag{1.34}$$

where f_b is the bending stiffness, m_b is penalty factor for non-monotonic angle changes, p_b is penalty factor for angles going to zero, $\Delta\phi$ is the change of the angle, and $\Delta\phi^*$ is non-negative measure for non-monotony of the change of the angle. The purpose of all the penalty factors is to prevent intermediate chaotic shapes. Such shapes are obtained if similar parts between the initial and the target shape are not maintained during the metamorphosis.

The total energy E_{total} needed for the shape metamorphosis from $S^I = (P_0^I, \dots, P_{n_I}^I)$ to $S^T = (P_0^T, \dots, P_{n_T}^T)$ is the sum of the sums over all bending energies and stretching energies:

$$E_{total}(S^I, S^T) = \sum_{\{segmentpairs\}} E_s(L_I, L_T) + \sum_{\{pointpairs\}} E_b(\phi), \tag{1.35}$$

where $\{segmentpairs\}$ is the set of all corresponding pairs of line segments in the initial and the target shape, and $\{pointpairs\}$ is the set of all point pairs between the corresponding segment points. The degree of morphing, which is the matching measure used in the recognition, is defined in the following way:

$$E_a(S^I, S^T) = E_{total}(S^I, S^T) + f * |n_I - n_T|, \tag{1.36}$$

where the factor f emphasizes the fact that similar shapes tend to have same number of segmentation points. Its value is found experimentally for each alphabet or other symbol set used.

The method was tested for one hundred cursive words and seven hand-drawn line figures written by twelve subjects. For each reference sample were four test samples. Recognition results for the writer-dependent tasks varied from 89.7% up to 99.0%.

1.5.3 Structural and syntactical methods

Structural and syntactical methods are used for various pattern recognition tasks in which the structure of the pattern is paramount and can be used for classification. The existence or absence of certain features in a patter does not always contain enough information for its classification – also some information on the relations of the features is needed.

In syntactical methods the features found in patterns are considered as primitives of a formal language. The language consist of grammatic rules which define the possible relations of the features, or in other words, define the structure of the pattern. The very basic features are represented by the terminal symbols of the language. The terminal symbols form the nonterminal symbols of the language according to the grammar. The recognition of the patterns is done parsing. Each class has its own grammatic rules and the pattern is classified to the syntactically corresponding class. The grammatic rules can be presented in two ways: as a string grammar or a graphical grammar. A string grammar defines in an one-dimensional way how a string of nonterminal symbols can be replaced by terminal symbols. The graphical grammars can present the relationships of the features in a higher dimensional way (Schalkoff 1992).

Example 11: A good and simple example of a syntactical method is Picture Description Language (PDL) introduced by A. C. Shawn in 1969. The primitives of PDL are straight line segments with heads and tails, four directions, and three lengths. The grammatic rules describe how the line segments can be connected to each other from their heads or tails. The grammatic rules can be presented with a tree in which the leaves correspond to the strokes and the nodes to their connections (Wang and Gupta 1991).

Structural methods are also based on an analysis of the features and their relations. They differ from the syntactical methods as the classification is not based on parsing but on matching and various decision rules. The recognition can be done hierarchically: the candidate classes are first ruled out or selected according to the low-level and then to the more complicated high-level features and structures. The pattern is classified to the class with matching structure. In case of an ambiguity decisions rules containing context and language information or rules specific to the confusing class pairs can be used. The hierarchical matching and the decision procedures can usually be described with decision trees. Two structural methods designed for on-line recognition of isolated characters are described in the following examples.

Example 12: Chan and Yeung (1998) have proposed a structural method in which the structure of the unknown pattern can be deformed if it does not directly match to any of the structures found in the model base. The method is called Elastic Structural Matching. The structures found in the characters are: a line segment, a counter-clockwise and a clockwise curve, a loop, and a dot. The structures, excluding a loop and a dot, are associated with a quantified direction from the starting point to the end point of the corresponding segment. A character is composed of a sequence of the structure types and their possible directions. The order of the elements of the sequence corresponds to the order in which they have been written.

The recognition is done on the basis of the first matching model found in the model base. At first the original pattern structure is matched against all the structures of the models. If no matching model is found, the structure of the pattern is modified. The structure is modified in three levels and the matching is repeated after each level. In the first level the primitive types are modified: a line may be changed to either one of the curves, and similarly both curves can be replaced with a line. If still no match is found, the directions of the original structures are modified: a direction can be changed to its neighborings directions. In the last level, both the structure types and their directions are modified simultaneously.

This method was tested with alphanumerics: digits and both lower and upper case letters. The recognition rates for multi-writer data were 98.60%, 98.49%, 97.44%, and 97.40% for digits, lower and upper case letters, and all characters together, respectively. The data was written by 150 subjects and it contained one sample of each class from all the writers.

Example 13: An hierarchical method based on structural features of three levels, binary decision trees, and matching has been described by Kerrick and Bovik (1988). First the low-level features of the pattern to be recognized are examined. They are easy to extract, and as they contain both local and global structural information on the character they are very effective in the pruning of the possible matching classes. The low-level features are: the existence endpoints of the strokes and the tees, and their locations in the bounding box of the character. The tees are points where two strokes are touching each other. The location can have only four values as it presents one of the quadrants of the bounding box. Next the intermediate-level features, the corners and their locations, are found by a chain code based algorithm developed by Freeman and Davis (1977). A corner contains more global information, as it is less locally defined, than an endpoint or a tee.

In the first stage of recognition the absence or the presence of particular low- and intermediate-level features in given quadrants are tested according to a binary decission tree. In the next stage the recognition is done by matching the unknown pattern against the remaining candidate set. The matching is done on stroke basis so the candidate set is further reduced by a parsing algorithm which tries to establish a stroke-to-stroke correspondence between two patterns. In the matching stage high-level features are used. Each stroke is associated with a shape, which can be either straight or curved. A straight stroke can have four possible orientations. The curved strokes are divided to four segments of equal length. An average chain code value is computed for each segment and thus the shape can be coded with four digits. Two strokes are considered to be similar if they both are straight and have same orientation, or the digits of their shape codes are same or differ only by one. To handle the ambiquity of several remaining candidates after the matching also the aspect ratios of the characters are compared and some additional class dependent feature tests are performed.

1.6 Aims of this research

Chapter 2

Data

2.0.1 Databases

2.0.2 Preprocessings

2.0.3 Normalizations

Chapter 3

Distances

3.0.4 Dynamic Time Warping

3.0.5 Point to point distance

3.0.6 Point to line distance

3.0.7 Simple area

3.0.8 A kind of area

Chapter 4

Clustering

4.0.9 Requirements for prototypes

4.0.10 Semiautomatic clustering method

4.0.11 Problems

Chapter 5

Recognition

5.0.12 kNN

5.0.13 Prototype pruning and matching order

Chapter 6

Adaptation

6.0.14 Adaptive systems

6.0.15 User's adaptation

6.0.16 System's adaptation

6.0.16.1 Learning Vector Quantification

6.0.16.2 Prototype adding

6.0.16.3 Hybrid rule

6.0.16.4 Prototype inactivation

6.0.16.5 Committee classifier

Chapter 7

Human experiment

Chapter 8

Results

8.0.17 The effects of preprocessings and normalizations

8.0.18 The effects of distances

8.0.19 The effects of adaptation

Chapter 9

Conclusions

9.0.20 Contributions

9.0.21 Future work

9.0.22 Conclusions

References

Arakawa, H. (1983). On-line recognition of handwritten characters - alphanumerics, hiragana, katakana, kanji. *Pattern Recognition 16*(1), 9–16.

Bellegarda, E. J., J. R. Bellegarda, D. Nahamoo, and K. S. Nathan (1994). A fast statistical mixture algorithm for on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*(12), 1227–1233.

Bellegarda, E. J., J. R. Bellegarda, D. Nahamoo, and K. S. Nathan (1995). A discrete parameter hmm approach to on-line handwriting recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 2631–2622. IEEE.

Brault, J.-J. and R. Plamondon (1993). Segmenting handwritten signatures at their perceptually important points. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*(9), 953–957.

Chan, K.-F. and D.-Y. Yeung (1998, August). A simple yet robust structural approach for recognizing on-line handwritten alphanumerical characters. In *Proceedings of the sixth international workshop on frontiers in handwriting recognition*, pp. 229–238.

Comerford, R. (1998, May). Pocket computers ignite os battle. *IEEE Spectrum*, 43–48.

Frankish, C., R. Hull, and P. Morgan (1995). Recognition accurancy and user acceptance of pen interfaces. In *Proceedings ACM CHI'95 Conference on Human Factors in Computing Systems*.

Freeman, H. and L. S. Davis (1977). A corner-finding algorithm for chain-coded curves. *IEEE Transactions on Computers*, 287–303.

Fujisaki, T., T. E. Chefalas, J. Kim, C. C. Tappert, and C. G. Wolf (1991). On-line run-on character recognizer: Desing and performance. *International Journal of Pattern Recognition and Artificial Intelligence 5*(1 and 2), 123–137.

Gardner, W. D. (1997). Putting pen to smart paper. *IBM Research magazine* (3).

Govindan, V. K. (1990). Character recognition - a review. *Pattern Recognition 23*(7), 671–683.

Guyon, I. and P. S. P. Wang (Eds.) (1993, August). *International journal of pattern recognition and artificial intelligence, Special issue: Advances in pattern recognition systems using neural networks*, Volume 7.

Guyon, I. and C. Warwick (1996). Joint ec-us survey of the state-of-the-art in human language technology. url=http://www.cse.ogi.edu/CSLU/HLTsurvey.htm/.

Hu, J., M. K. Brown, and W. Turin (1996). Hmm based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(10), 1039–1045.

Hu, J., S. G. Lim, and M. K. Brown (1998, August). Hmm based writer independent on-line recognition handwritten character and word recognition. In *Proceedings of the sixth international workshop on frontiers in handwriting recognition*, pp. 143–155.

Kerrick, D. D. and A. C. Bovik (1988). Microprocessor-based recognition of handprinted characters from a tablet input. *Pattern Recognition 21*(5), 525–537.

Kuklinski, T. T. (1984). Components of handprint style variability. In *7 th International Conference on Pettern Recognition*, pp. 924–926.

Li, X. and D.-Y. Yeung (1997). On-line handwritten alphanumeric character recognition using dominant points in strokes. *Pattern Recognition 30*(1), 31–44.

Loy, W. and L. Landay (1982, July). An on-line procedure for recognition of handprinted alphanumeric characters. *Pattern Recognition 4*(4), 422–427.

Martens, R. and L. Claesen (1997). Dynamic programminng optimization for on-line signature verification. In *Proceeding of International Conference on Document Analysis and Recognition*, pp. 653–656. IEEE.

Mel, M. W., S. M. Omohundro, A. D. Robinson, S. S. Skiena, K. H. Thearling, L. T. Young, and S. Wolfram (1988, June). Tablet: personal computer in the year 2000. *Communications of the ACM 31*(6), 639–648.

Nathan, K., J. R. Bellegarda, D. Nahamoo, and E. J. Bellegarda (1993). On-line handwriting recognition using continuous parameter hidden markov models. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Volume 5, pp. 121–124.

Neisser, U. and P. Weene (1960). A note on human recognition of hand-printed characters. *Information and Control* (3), 191–196.

Nouboud, F. and R. Plamondon (1990). On-line recognition of handprinted characters: survey and beta tests. *Pattern Recognition 23*(9), 1031–1044.

Nouboud, F. and R. Plamondon (1991). A sructural approach to on-line character recognition: System design and applications. *International Journal of Pattern Recognition and Artificial Intelligence 5*(12), 311–335.

Odaka, K., H. Arakawa, and I. Masuda (1982). On-line recognition of handwritten characters by approximating each stroke with several points. *IEEE Transactions on Systems, Man and Cybernetics 12*(6), 898–903.

Parizeau, M. and R. Plamondon (1994). Machine vs humans in a cursive script reading experiment without linguistic knowledge. In *Proceedings of International Conference on Pattern Recognition*, Volume 2, pp. 93–98.

Pavlidis, I., R. Singh, and N. P. Papanikolopoulos (1997). An on-line handwritten note recognition method using shape metamorphosis. In *International Conference on Document Analysis and Recognition*, Volume 2, pp. 914–918. IEEE.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 267–295.

Rigoll, G., A. Kosmala, J. Rottland, and C. Neukirchen (1996). A comparison between continuous and discrete density hidden markov models for cursive handwriting recognition. In *Proceedings of International Conference on Pattern Recognition*, Volume 2, pp. 205–209. IEEE.

Sankoff, D. and J. B. Kruskal (1983). *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Addison-Wesley.

Sato, Y. and K. Kogure (1982). Online signature verification based on shape, motion, and writing pressure. In *Proceedings of International Conference on Pattern Recognition*, Volume 2, pp. 823–826. IEEE.

Scattolin, P. (1995, November). Recognition of handwritten numerals using elastic matching. Master's thesis, Concordia University, Canada.

Schalkoff, R. (1992). *Pattern recognition: statistical, structural and neural approaches.* John Wiley Sons, Inc.

Sederberg, T. and E. Greenwood (1992). A physically besed approach to 2-d shape blending. *Computer Graphics 26*(2), 25–34.

Srihari, S. N. (1985). *Computer text recognition and error correction*, Chapter Introduction, pp. 1–4. IEEE Computer Society Press.

Starner, T., R. S. J. Makhoul, and G. Chou (1994). On-line cursive handwriting recognition using speech recognition methods. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Volume 5, pp. 125–128. IEEE.

Suen, C. Y. (1979, November). A study on man-machine interaction problems in character recognition. *IEEE Transactions on Systems, Man, and Cybernetics 9*(11), 732–737.

Tappert, C., C. Y. Suen, and T. Wakahara (1990). The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*(8), 787–808.

Tappert, C. C. (1984). Adaptive on-line handwriting recognition. In *International Conference on Pattern Recognition*, pp. 1004–1007. IEEE.

Veltman, S. R. and R. Prasad (1994). Hidden markov models applied to on-line handwritten isolated character recognition. *IEEE Transactions on Image Processing 3*(3), 314–318.

Wakahara, T. and K. Odaka (1997, December). On-line cursive kanji character recognition using stroke-based affine transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*(12), 1381–1385.

Wang, P. S.-P. and A. Gupta (1991). An improved structural approach for automated recognition of handprinted character. *International Journal of Pattern Recognition and Artificial Intelligence 5*(12), 97–121.

Ward, J. R. and B. Blesser (1985, September). Interactive recognition of handprinted characters for computer input. *IEEE Computer Graphics and Applications 9*(5), 24–37.

Ward, J. R. and T. Kuklinski (1988, May/June). A model for variability effects in handprinting with implications for design of handwriting character recognition systems. *IEEE Transactions on systems, Man, and Cybernetics 18*(3), 438–451.

Wilfonng, G., F. Sinden, and L. Ruedisueli (1996). On-line recognition of handwritten symbols. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(9), 935–940.

Wirtz, B. (1998, March). Method for the dynamic verification of an autograph character string on the basis of a reference autograph character string. United States Patent, patent number 5,730,468.

Yuen, H. (1996). A chain coding approach for real-time recognition of on-line handwritten characters. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Volume 6, pp. 3426–3429. IEEE.